



UNIVERSIDAD CARLOS III DE MADRID

Grado en Ingeniería Informática

Diseño e implementación de un sistema domótico basado en Raspberry Pi

TRABAJO FIN DE GRADO

Madrid, Febrero 2017

Autor: Héctor Santos Senra

Tutor: Javier Fernández Muñoz

Título: Diseño e implementación de un sistema domoticos basado en Raspberry Pi

Autor: Héctor Santos Senra

Director:

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Resumen

La temática del presente proyecto es el análisis, diseño e implementación de un sistema domótico instalado en un modelo físico de vivienda, que permitirá realizar tareas de monitorización y manejo de dispositivos de forma remota.

Para llevar a cabo estas tareas serán necesarios ciertos elementos hardware como sensores, actuadores, micro-controladores o fuentes de alimentación. Y como no podía ser de otra forma, el software también juega un papel muy importante en este proyecto, permitiendo almacenar y visualizar la información mediante un servidor web y una base de datos.

Las ventajas del sistema domótico desarrollado son varias. En primer lugar, se podrá usar desde cualquier dispositivo (pc, laptop, tablet, smartphone, ...) conectado a internet. Además el hecho de utilizar una plataforma de desarrollo libre aporta la versatilidad y comodidad a la hora de desarrollar, necesarias para llevar a cabo las tareas de monitorización y control que se pretenden implementar.

Palabras clave: Linux, Raspbian, RaspberryPi, Python, PHP, MySQL, JavaScript, jQuery

Indice de contenidos

Resumen	3
Resumen	11
1. Introducción	12
1.1 Motivaciones	12
1.2 Objetivos	12
1.3 Estructura del documento	13
2. Estado del arte	14
2.1 Introducción	14
2.2 Servicios en sistemas de domótica	15
2.2.1 Ahorro energético	15
2.2.2 Accesibilidad	15
2.2.3 Seguridad	16
2.2.4 Confort	16
2.2.5 Comunicaciones.....	16
2.3 El sistema domotico	18
2.3.1 Elementos	18
2.3.2 Arquitectura	18
2.4 RaspberryPi	19
2.5.1 Características	20
2.5.2 Distribuciones	21
2.5 Aplicaciones y software existente	23
2.6 Comparativa plataformas hardware.....	24
3. Gestión del proyecto	26
3.1 Introducción	26
3.2 Metodología del proyecto	26
3.2.1 Fases de la metodología adoptada	27
3.3 Ciclo de vida	29
3.4 Planificación	30
3.5 Presupuesto	32
3.5.1 Resumen de tiempo dedicado	32
3.5.2 Desglose: Coste de personal	32
3.5.3 Desglose: Coste de hardware	33
3.5.4 Desglose: Coste de software.....	35

3.5.5	Resumen de costes	36
4.	Análisis del sistema	37
4.1	Introducción	37
4.2	Casos de Uso	37
4.3	Diagramas UML	43
4.4	Requisitos del Sistema.....	45
4.4.1	Requisitos de Usuario	45
4.4.1.1	Requisitos de capacidad	46
4.4.1.2	Requisitos de restricción.....	49
4.4.2	Requisitos del Software	50
4.4.2.1	Requisitos funcionales	50
4.4.2.2	Requisitos no funcionales.....	55
4.5	Matriz de trazabilidad.....	57
5.	Descripción del sistema domótico.....	59
5.1	Introducción	59
5.2	Requisitos del sistema domótico	60
	Distribución de elementos	62
5.3	Infraestructura del sistema	68
	Servidor domotico	68
	Bus de comunicaciones	69
	Fuente de alimentación del sistema.....	69
	Sensores y actuadores.....	70
6.	Diseño del sistema	74
6.1	Lenguajes de programación	74
6.1.1	El servidor web	74
6.1.2	Aplicación web.....	76
6.2	Arquitectura del Sistema	77
6.3	Estructura de la Base de Datos	80
6.4	Diseño UML del sistema	82
7.	Implementación del sistema	85
7.1	Introducción	85
7.2	Herramientas de desarrollo del sistema.....	85
8.	Pruebas del sistema	88
8.1	Introducción	88
8.2	Procedimientos de pruebas.....	88

8.2.1 Pruebas de infraestructura hardware	89
8.2.2 Pruebas de funcionalidad	91
8.2.3 Matriz de trazabilidad.....	94
9. Conclusiones y líneas futuras	95
9.1 Conclusiones.....	95
9.2 Líneas futuras de mejora	95
9.3 Glosario de términos	96
10. Referencias.....	97
11. Anexos.....	98
11.1 ANEXO I Esquemas eléctricos	98
11.2 ANEXO II Manual de usuario.....	101

Indice de figuras

Figura 1. Servicios domótica.....	17
Figura 2. Logotipo Raspberry Pi.....	19
Figura 3. Risc OS Pi	21
Figura 4. Arch Linux ARM	22
Figura 5. Raspbian	22
Figura 6. Openhab.....	23
Figura 7. Open Domo	24
Figura 8. Modelo de calidad Metrica 3.....	27
Figura 9. Ciclo de vida.....	29
Figura 10. Diagrama de Gantt	31
Figura 11. Casos de uso CU-1, CU-2 Y CU-3	43
Figura 12. Casos de uso CU-7 y CU-8.....	44
Figura 13. Casos de uso CU-4 y CU-6.....	44
Figura 14. Icono temperatura	60
Figura 15. Icono presencia	61
Figura 16. Icono seguridad	61
Figura 17. Icono iluminación	62
Figura 18. Representación interfaz salón	62
Figura 19. Representación interfaz hall	63
Figura 20. Representación interfaz habitación.....	63
Figura 21. Representación interfaz cuarto de baño	64
Figura 22. Representación interfaz cocina	64
Figura 23. Representación interfaz garaje.....	65
Figura 24. Modelo físico vivienda.....	66
Figura 25. Bus GPIO	69
Figura 26. Conector bus GPIO	69
Figura 27. Fuente de alimentación	69
Figura 28. Sensor DHT-11	70
Figura 29. Sensor PIR.....	71
Figura 30. Sensor LDR.....	72
Figura 31. Motor DC.....	72
Figura 32. Montaje final modelo físico	73
Figura 33. Diagrama sistema hardware	73
Figura 34. Ejemplo comunicación usuario servidor web	74
Figura 35. Servidor LAMP	75
Figura 36. Framework Bootstrap.....	75
Figura 37. Lenguaje PHP	76
Figura 38. Lenguaje Python	76
Figura 39. Esquema MVC	78
Figura 40. Arquitectura Sistema Domótico	79
Figura 41. Diagrama Entidad-Relación	80
Figura 42. Diagrama Base de datos	81
Figura 43. Diagrama de secuencia. Validar mensajes.....	82
Figura 44. Diagrama de secuencia. Obtener temperatura	83
Figura 45. Diagrama de secuencia. Modificar actuador	84

Figura 46. Editor Sublime Text	85
Figura 47. Logo Winscp	85
Figura 48. Logo PUTTY	86
Figura 49. Logo Google Charts.....	87
Figura 50. Logo Fritzting	87
Figura 51. Logo MySQL Workbench	87
Figura 52. Esquema eléctrico motores	98
Figura 53. Esquema eléctrico sensores	99
Figura 54. Esquema eléctrico leds	100
Figura 55. Pantalla login aplicación	101
Figura 56. Pantalla menú principal	102
Figura 57. Pantalla selección monitorización ambiental	102
Figura 58. Pantalla monitorización ambiental cuarto de baño	102
Figura 59. Pantalla control seguridad garaje	102
Figura 60. Pantalla datos históricos.....	102

Indice de tablas

Tabla 1.....	20
Tabla 2. Comparativa plataformas hardware	25
Tabla 3. Detalle tareas.....	30
Tabla 4. Desglose del coste de personal.....	33
Tabla 5. Desglose del coste de Hardware.....	35
Tabla 6. Desglose del coste de Software	36
Tabla 7. Desglose del coste total	36
Tabla 8. Detalle Caso de uso.....	38
Tabla 9. Caso de uso 01.....	38
Tabla 10. Caso de uso 02	39
Tabla 11. Caso de uso 03	39
Tabla 12. Caso de uso 04	40
Tabla 13. Caso de uso 05	40
Tabla 14. Caso de uso 06	41
Tabla 15. Caso de uso 07	41
Tabla 16. Caso de uso 08	42
Tabla 17 Caso de uso 09	42
Tabla 18. Detalle Requisitos de usuario	46
Tabla 19. Requisito de capacidad visualizar aplicación	46
Tabla 20. Requisito de capacidad identificar usuario	46
Tabla 21. Requisito de capacidad leer valor instantáneo	47
Tabla 22. Requisito de capacidad leer valor medio	47
Tabla 23. Requisito de capacidad consultar histórico	47
Tabla 24. Requisito de capacidad controlar actuadores.....	48
Tabla 25. Requisito de capacidad visualizar estado actuadores	48
Tabla 26. Requisito de restricción limitar uso internet.....	49
Tabla 27. Requisito de restricción limitar plazo consulta	49
Tabla 28. Detalle Requisito de software.....	50
Tabla 29. Requisito funcional dispositivo telemático	50
Tabla 30. Requisito funcional dispositivo con conexión	51
Tabla 31. Requisito funcional crear usuarios.....	51
Tabla 32. Requisito funcional identificar accesos	51
Tabla 33. Requisito funcional control accesos.....	52
Tabla 34. Requisito funcional leer dato sensor	52
Tabla 35. Requisito funcional mostrar dato	52
Tabla 36. Requisito funcional calculo valor medio	53
Tabla 37. Requisito funcional almacenar valor medio.....	53
Tabla 38. Requisito funcional almacenar valores	53
Tabla 39. Requisito funcional mostrar historico.....	54
Tabla 40. Requisito funcional realizar consultas.....	54
Tabla 41. Requisito funcional modificar actuadores	54
Tabla 42. Requisito funcional consultar estado actuadores	55
Tabla 43. Requisito funcional almacenar acciones	55
Tabla 44. Requisito restricción conexión internet	55

Tabla 45. Requisito restricción plazo consulta	56
Tabla 46. Requisito restricción crear usuarios.....	56
Tabla 47. Requisito restricción valor medio luminosidad	56
Tabla 48. Matriz de trazabilidad. Requisitos de capacidad con requisitos funcionales	58
Tabla 49. Modelo temperatura	60
Tabla 50. Modelo presencia	61
Tabla 51. Modelo seguridad.....	61
Tabla 52. Modelo iluminación.....	62
Tabla 53. Detalle elementos modelo físico	67
Tabla 54. Características sensor DHT-11	70
Tabla 55. Características sensor PIR	71
Tabla 56. Características motor dc	72
Tabla 57. Detalle prueba del sistema	89
Tabla 58. Prueba Hardware PHW-01.....	89
Tabla 59. Prueba Hardware PHW-02.....	89
Tabla 60. Prueba Hardware PHW-03.....	90
Tabla 61. Prueba Hardware PHW-04.....	90
Tabla 62. Prueba Hardware PHW-05.....	90
Tabla 63. Prueba Hardware PHW-06.....	91
Tabla 64. Prueba Funcionalidad PAW-01	91
Tabla 65. Prueba Funcionalidad PAW-02	91
Tabla 66. Prueba Funcionalidad PAW-03	92
Tabla 67. Prueba Funcionalidad PAW-04	92
Tabla 68. Prueba Funcionalidad PAW-05	92
Tabla 69. Prueba Funcionalidad PAW-06	93
Tabla 70. Prueba Funcionalidad PAW-07	93
Tabla 71. Matriz de trazabilidad. Requisitos funcionales con pruebas de funcionalidad	94

Resumen

La temática del presente proyecto es el análisis, diseño e implementación de un sistema domótico instalado en un modelo físico de vivienda, que permitirá realizar tareas de monitorización y manejo de dispositivos de forma remota.

Para llevar a cabo estas tareas serán necesarios ciertos elementos hardware como sensores, actuadores, micro-controladores o fuentes de alimentación. Y como no podía ser de otra forma, el software también juega un papel muy importante en este proyecto, permitiendo almacenar y visualizar la información mediante un servidor web y una base de datos.

Las ventajas del sistema domótico desarrollado son varias. En primer lugar, se podrá usar desde cualquier dispositivo (pc, laptop, tablet, smartphone, ...) conectado a internet. Además el hecho de utilizar una plataforma de desarrollo libre aporta la versatilidad y comodidad a la hora de desarrollar, necesarias para llevar a cabo las tareas de monitorización y control que se pretenden implementar.

Palabras clave: Linux, Raspbian, RaspberryPi, Python, PHP, MySQL, JavaScript, jQuery

1. Introducción

En esta primera sección del documento se describen las motivaciones y objetivos que han marcado todo el desarrollo del presente Trabajo Fin de Grado, a partir de ahora, TFG. También se detallará la estructura general del proyecto y las fases en las que se ha dividido.

1.1 Motivaciones

Desde hace algunos años las plataformas hardware opensource han revolucionado el sector del software, convirtiendo tareas del ámbito industrial, como el sensado o la monitorización de sistemas, en tecnologías de uso masivo que reúnen los requisitos para que la imaginación sea el único límite a la hora de diseñar ingeniosos sistemas.

Construir un robot asistente o montar una estación meteorológica son proyectos abordables mediante el uso de estas herramientas. Proyectos tan dispares y complejos se simplifican de forma sustancial, acercando gran número de nuevos programadores y usuarios finales a la tecnología.

La integración de estas plataformas hardware en sistemas operativos habituales, como es el caso de Windows 10 y su edición especial dedicada a IoT, son cada vez mas habituales. De esta forma se mejora la comunicación entre dispositivos distintos, así como la programación y visualización de datos.

1.2 Objetivos

El objetivo principal consiste en el desarrollo de un sistema domótico de bajo coste, basado en código libre (UNIX/LINUX), a partir de ahora, opensource. Dicho sistema permitirá al usuario manejar de forma fácil y sencilla elementos finales como motores y leds

Para llevar a cabo estos objetivos serán necesarios determinados elementos hardware/software, así como el modelo físico de la vivienda que permita comprobar el correcto funcionamiento del sistema completo.

El elemento software principal será el sistema operativo RaspBian (basado en Debian) instalado en el microcontrolador RaspberryPi. El segundo elemento software importante será el servidor web Apache que gestionará la correspondiente aplicación web desarrollada bajo el patrón de desarrollo Modelo-Vista-Controlador (MVC)

El sistema domótico estará compuesto de varios sub-sistemas: la aplicación web, la base de datos y el conjunto de sensores y actuadores.

La aplicación web se ha desarrollado sobre un servidor LAMP que será accesible desde el exterior y que actuará de interfaz entre el usuario del sistema domótico y el propio sistema.

Algunas de las acciones que podrá realizar el usuario son:

- Consultar temperatura y humedad de una habitación.
- Controlar apertura puertas/persianas
- Consultar histórico de temperaturas y humedades
- Consultar movimiento en una habitación
- Controlar iluminación interior

1.3 Estructura del documento

En este apartado se detallan los diferentes puntos en los que se ha estructurado el documento, dando así una visión global del mismo.

- En el presente capítulo 1, se han introducido las motivaciones, objetivos y detalles del documento.
- En el capítulo 2, se analiza el estado de la cuestión, analizando la evolución de la tecnología en el sector domótico y comparando las tecnologías o productos actuales en relación a las plataformas hardware/software opensource.
- En el capítulo 3, se detalla la planificación que se quiere seguir a lo largo de las distintas fases del proyecto, así como el presupuesto del sistema completo de acuerdo con la duración estimada.
- En el capítulo 4, se describen los procesos de análisis de requisitos, que definen las necesidades que el sistema debe solucionar, de decisiones de diseño tomadas para el desarrollo de interfaces y de implementación.
- En el capítulo 5, se describe el sistema domotico, detallando los elementos que lo componen y la distribución de los mismos a lo largo del modelo físico de vivienda.
- En el capítulo 6, se describe el proceso llevado a cabo para diseñar el sistema, así como la estructura de la base de datos.
- En el capítulo 7, se describe la implementación sistema domotico.
- En el capítulo 8, se detalla la batería de pruebas realizada sobre el sistema domotico.

2. Estado del arte

En este apartado, se presentará la situación actual del sector domotico y de las plataformas de hardware/software libre, en general. También se especificarán los servicios que pueden aportar los sistemas domoticos, así como una descripción detallada de algunos de los sistemas domóticos actuales.

2.1 Introducción

Según la Real Academia de la Lengua Española, domótico o domótica se refiere al conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda.

El término deriva de la unión de la palabra ‘domus’ (que quiere decir, casa, en latín) y tica (de automática, palabra que en latín quiere decir ‘que funciona por sí sola’). La domótica por tanto puede entenderse como el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permiten una gestión eficiente del uso de la energía, aportan seguridad y confort, además de comunicación entre el usuario y el sistema.

Dada la morfología del presente proyecto (sistema domotico conectado a internet), se pretende mejorar de forma clara el ámbito de las comunicaciones entre el usuario y el sistema domotico, así como la capacidad de ampliación, configuración e instalación de nuevos elementos en el sistema desarrollado.

El control automático de las viviendas lleva en auge desde principios de los años ochenta, en los que los sistemas disponibles hacían uso de la tecnología telefónica mediante cable de cobre. Estos sistemas permitían un control bastante limitado en cuanto a la gestión y modificación de los propios sistemas.

En la actualidad el sector domótico se encuentra repartido en, al menos, otros dos sectores: el sector industrial (HoneyWell, Siemens, etc.) y la misma industria del software (Microsoft IoT). Estos gigantes, a pesar de tener un régimen privado, están adoptando cada vez mas el uso de tecnologías abiertas que permiten un aprendizaje estandarizado y con comunidades de desarrolladores en constante crecimiento.

Gracias a la expansión del software libre (o código abierto) y sus importantes ventajas respecto a los sistemas privativos, un usuario final es capaz de construir su propio sistema domótico de bajo coste y 100% funcional.

2.2 Servicios en sistemas de domótica

En la actualidad, la domótica aporta soluciones dirigidas a todo tipo de viviendas y gracias a la evolución tecnológica permite mejorar numerosas situaciones cotidianas en lo que a calidad de vida dentro del hogar se refiere. A continuación se enumerarán los servicios típicos que puede introducir o mejorar un sistema domotico actual:

2.2.1 Ahorro energético

Al gestionar de forma inteligente la iluminación, la climatización, el agua caliente sanitaria, el riego, los electrodomésticos, etc., se consigue un mejor aprovechamiento de los recursos naturales. Haciendo uso de las tarifas horarias de menor coste, y reduciendo así, la factura energética. Además, mediante la monitorización de consumos, se obtiene la información necesaria para modificar los hábitos y aumentar el ahorro y la eficiencia.

Elementos importantes:

- Climatización: Programación y zonificación.
- Gestión eléctrica:
 - Racionalización de cargas eléctricas: Desconexión de equipos de uso no prioritario en función de cierto consumo eléctrico en cierto tiempo.
 - Gestión de tarifas: Derivando el funcionamiento de los aparatos de mayor consumo energético a ciertas horas del día en el cual la tarifa es más barata.
- Uso de energías renovables.

2.2.2 Accesibilidad

Se trata de facilitar el manejo de los elementos del hogar a las personas con discapacidades de la forma que más se ajuste a sus necesidades, además de ofrecer servicios de tele-asistencia para aquellos que lo necesiten.

El concepto "diseño" para todos es un movimiento que pretende crear la sensibilidad necesaria para que al diseñar un producto o servicio se tengan en cuenta las necesidades de todos los posibles usuarios, incluyendo las personas con diferentes capacidades o discapacidades, es decir, favorecer un diseño accesible para la diversidad humana.

2.2.3 Seguridad

Consiste en una red de seguridad encargada de proteger tanto los bienes patrimoniales como la seguridad personal.

- Alarmas de intrusión (Anti-intrusión): Se utilizan para detectar o prevenir la presencia de personas extrañas en una vivienda o edificio.
 - Detección de un posible intruso (Detectores volumétricos o perimetrales)
 - Cierre de persianas puntual y seguro
 - Simulación de presencia
- Alarmas de detección de incendios, fugas de gas, escapes de agua, concentración de monóxido en garajes cuando se usan vehículos de combustión.
- Alerta médica, tele asistencia. ☑ Acceso a cámaras IP.

2.2.4 Confort

El confort, conlleva todas las actuaciones que se puedan llevar a cabo que mejoren las condiciones de comodidad de una vivienda. Dichas actuaciones, a través de la gestión de dispositivos y actividades domésticas, permiten abrir, cerrar, apagar, encender, regular... los electrodomésticos, la climatización, ventilación, iluminación natural y artificial, persianas, toldos, puertas, cortinas, riego, suministro de agua, gas, electricidad...).

Estas actuaciones pueden ser de carácter activo, pasivo o mixto:

- Iluminación:
 - Apagado general de todas las luces de la vivienda.
 - Automatización del apagado o encendido en cada foco de luz.
 - Regulación de la iluminación según necesidades ambiente.
- Automatización: Los distintos sistemas/instalaciones/equipos dotándolos de un control eficiente y de fácil manejo.
- Integración del portero al teléfono o del video portero al televisor.
- Control vía internet.
- Gestión Multimedia del ocio electrónico.
- Generación de programas de forma sencilla para el usuario.

2.2.5 Comunicaciones

Mediante el control y supervisión remoto de la vivienda a través de dispositivos inteligentes (pc, smartphone, tablet, ...) se permite la recepción de avisos de anomalías e información del funcionamiento de equipos e instalaciones.

La instalación domótica permite la transmisión de voz y datos, incluyendo textos, imágenes, sonidos (multimedia) con redes locales (LAN).

Servicios extra :

- compartiendo acceso a Internet;
- recursos e intercambio entre todos los dispositivos,
- acceso a nuevos servicios de telefonía IP,
- televisión digital, por cable,
- diagnóstico remoto,
- videoconferencias,
- tele-asistencia...



Figura 1. Servicios domótica

2.3 El sistema domotico

2.3.1 Elementos

Tradicionalmente los elementos que componen un sistema domótico, incluyen principalmente tres tipos:

- Controladores.
- Sensores.
- Actuadores.

2.3.2 Arquitectura

Desde el punto de vista donde reside la inteligencia del sistema de domótica, hay varias arquitecturas distintas.

- Arquitectura Centralizada: un controlador centralizado recibe información de múltiples sensores y, una vez procesada, genera las órdenes oportunas para los actuadores.
- Arquitectura Distribuida: toda la inteligencia del sistema está distribuida por todos los módulos sean sensores o actuadores. Suele ser típico de los sistemas de cableado en bus, o redes inalámbricas.
- Arquitectura mixta: sistemas con arquitectura descentralizada en cuanto a que disponen de varios pequeños dispositivos capaces de adquirir y procesar la información de múltiples sensores y transmitirlos al resto de dispositivos distribuidos por la vivienda, p.ej. aquellos sistemas basados en Zigbee y totalmente inalámbricos.

2.4 RaspberryPi

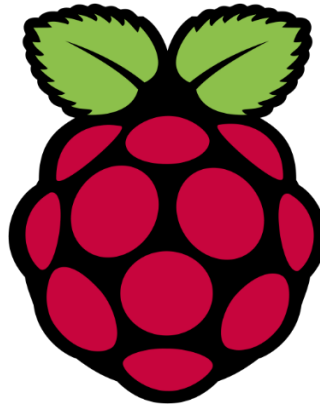


Figura 2. Logotipo Raspberry Pi

La Raspberry Pi es un ordenador con el tamaño de una tarjeta de crédito, también llamado placa computadora (SBC) y de muy bajo coste, es un proyecto que comenzó en 2006, en la universidad de Cambridge para fomentar la enseñanza de las ciencias computacionales en los colegios, en 2012 se comenzó a fabricar y comercializar en serie.

El sistema operativo que utiliza es Linux y dispone de una gran cantidad de distribuciones para ella, gracias a esto un usuario puede utilizarla como si de un PC con Linux se tratara, también dispone de una comunidad de usuarios y desarrolladores que crece sin parar, ya que aparte de su objetivo educacional, la Raspberry Pi también dispone de grandes características multimedia, en especial los últimos modelos, convirtiéndola en una plataforma de entretenimiento muy poderosa y a un precio muy competitivo.

La SBC (Single Board Computer) Raspberry Pi ha cosechado grandes éxitos por su bajo precio y por sus grandes posibilidades, tanto para los desarrolladores de software, estudiantes de informática, como para los aficionados que desean emplearla para sus propios proyectos caseros.

En los siguientes apartados se analizarán las características y distribuciones para Raspberry Pi más importantes.

2.5.1 Características

Raspberry PI 3 Modelo B 1Gb

System On A Chip (SoC)	Broadcom BCM2387
CPU	1,2 GHz de cuatro núcleos ARM Cortex-A53 (familia ARM11)
Juego de Instrucciones	RISC de 32 bits
GPU	<ul style="list-style-type: none"> Dual Core VideoCore IV[®] Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación. Capaz de 1 Gpixel / s, 1.5Gtexel / s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA
Memoria RAM	1GB LPDDR2.
Puertos USB 2.0	2 (via hub USB integrado)
Conector GPIO	<ul style="list-style-type: none"> 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira Proporcionar 27 pines GPIO, así como 3,3 V, +5 V y GND líneas de suministro
Conectividad red	10/100 Ethernet
Salida video	HDMI 1.4 1920x1200
Salida audio	Salida audio estéreo vía 3.5 mm TRRS jack
Almacenamiento	Tarjeta micro SD
Alimentación	5 V a 2A micro USB
Consumo	700 mA (3.5 W)

Tabla 1.

2.5.2 Distribuciones

Para Raspberry PI existen una gran cantidad de distribuciones, cada una con sus propias características y peculiaridades como por ejemplo OpenElec que está principalmente pensada para convertir nuestra Raspberry Pi en un completo centro multimedia. La mayoría de las ditribuciones son adaptaciones a la plataforma ARM de sus respectivas versiones de escritorio.

La distribución que cuenta con el apoyo oficial y se puede decir que es la más estable y optimizada a nivel general, es Raspbian OS que se basa en la distribución Debian Wheezy (Debian 7.0), esta será la elegida para este proyecto como resultado de la comparativa que se muestra a continuación.

RISC OS es uno de los pocos sistemas operativos no basados en Linux que existen para la Raspberry Pi. De hecho, RISC OS es un sistema operativo británico desarrollado por Acorn Computers (los creadores de ARM) y que se distribuye bajo licencia Open-Source.

Aunque su soporte y catálogo de aplicaciones disponibles no sea tan amplio como el de otras distribuciones, también está considerado como una de los sistemas operativos oficiales de la Raspberry Pi y es especialmente interesante en cuanto a que se ha creado en torno a la plataforma ARM desde cero.



Figura 3. Risc OS Pi

Arch Linux es otro de los grandes nombres en cuanto a distribuciones Linux. Se caracteriza por su simplicidad, elegancia, coherencia del código y minimalismo. Pero la simplicidad no quiere decir facilidad de uso, ya que Arch Linux es bastante conocida por ser poco amigable y recomendable solo para gente con conocimientos más elevados. Ahora Arch Linux soporta ARM y por tanto también puede instalarse en la Raspberry Pi y otras placas SBC similares.



Figura 4. Arch Linux ARM

Raspbian OS es la distribución por excelencia para la Raspberry Pi. Es la más completa y optimizada de las existentes, por eso cuenta con apoyo oficial. Raspbian OS se basa en la potente distro Debian Wheezy (Debian 7.0) optimizando el código de ésta para la SBC Raspberry Pi.

La distribución es ligera para moverse ágilmente en el hardware de la Raspberry Pi, comenzó con un entrono de escritorio LXDE y Midori como navegador web predeterminado, pero la Raspberry Pi Foundation ha creado un entorno de escritorio especial llamado PIXEL (Pi Improved Xwindows Environment Lightweight). Además incluye herramientas de desarrollo muy interesantes, como IDLE para Python, Scratch para programar videojuegos (muy interesante sobre todo si se combina con Arduino), la tienda de aplicaciones denominada Pi Store, etc...



Figura 5. Raspbian

2.5 Aplicaciones y software existente

openHAB nace para satisfacer las necesidades de los que quieren un hogar inteligente y barato. La domótica se extiende cada vez más y esta distribución Linux incluye funcionalidades para controlar los aparatos e iluminación de tu casa. Con una sencilla interfaz gráfica se puede gestionar todo el hogar de forma centralizada y cómoda gracias a que soporta tecnologías como KNX, Insteon, HomeMatic, etc...



Figura 6. Openhab

OpenDomo OS es un sistema operativo de código abierto especialmente diseñado para domótica. Con él podrás crear una vivienda totalmente domotizada, además de crear automatismos independientes, realizar proyectos para ahorro energético en tu hogar, etc... Por supuesto OpenDomo utiliza un kernel Linux, y por tanto es una distro.




OpenDomo OS está disponible tanto para Raspberry Pi como para ejecutarlo en una máquina virtual desde tu PC para practicar la domótica. También existe un proyecto de hardware embebido denominado ODNetwork que emplea OpenDomo OS como sistema operativo base, además de otros proyectos que también se sirven de OpenDomo OS para su funcionamiento, como DominoOSE, un firmware para Arduino Ethernet y ODControl, otro sistema empleado en domótica.



Figura 7. Open Domo

2.6 Comparativa plataformas hardware

En la TABLA xxx se expone un listado de distintas plataformas hardware con sus especificaciones técnicas.

Tipo placa	Descripción	Especificaciones	Imagen	Precio
BeagleBone Black	Es una plataforma de desarrollo apoyada por la comunidad para desarrolladores y aficionados.	AM335x 1GHz ARM® Cortex-A8 2GB on-board flash y lector microSD HDMI 2x 46 pin headers Ethernet		45,00 €
CubieBoard	Placa de desarrollo basado en Allwinner A10, con una amplia gama de opciones de IO. La placa ofrece SATA y un interfaz de perno extendido para el acceso de bajo nivel al SOC. "	1G ARM cortex-A8 processor, NEON, VFPv3, 256KB L2 cache 512M/1GB DDR3 @480MHz 96 extend pin including I2C, SPI, RGB/LVDS, CSI/TS, FM-IN, ADC, CVBS, VGA, SPDIF-OUT, R-TP. HDMI Ethernet Precio: \$39-\$56		48,00 €
Nanode	Placa basada en Arduino con conectividad web built-in. Es una plataforma de bajo costo para el desarrollo creativo de ideas	ATmega328P microcontroller Hasta 14 I/O digitales 32KB ISP memoria flash SPI memoria de expansion (SRAM, Flash o FRAM) Ethernet 868MHz 433MHz		59 €


	conectadas a la red. Es como un Arduino con un escudo Ethernet incorporado. Fácilmente actualizable a inalámbrico agregando un kit de radio de 433Mhz ".			
Hackberry	Basado en el popular 1.2Ghz ARM Allwinner A10, el Hackberry A10 es un poderoso hacker Android / Linux PC. El Hackberry A10 tiene WiFi y Ethernet	1.2GHz Allwinner A10 ARM Cortex A8 DDR3 512MB / 1GB 4GB NAND storage HDMI		115 €

Tabla 2. Comparativa plataformas hardware

3. Gestión del proyecto

En este capítulo se detalla la planificación que se seguirá a lo largo de las distintas fases del proyecto, así como el presupuesto del mismo. La planificación se acompaña de un diagrama de Gantt a modo de resumen gráfico del tiempo empleado en cada tarea. El presupuesto incluirá un desglose de todos los costes por separado para tener una mejor apreciación de los costes finales.

3.1 Introducción

Se pretende realizar una buena planificación del proyecto desde una primera instancia. Para ello, una buena estimación de tiempos en cada fase y sus tareas marcará el éxito de la planificación inicial.

Una mala estimación (no realista y sin valoraciones) de tiempos en tareas puede ser el punto de inflexión que determine si se cumplen los plazos o no a los que se compromete el equipo de desarrollo. Es decir, en muchos casos, la diferencia entre que el cliente termine satisfecho, o no.

Para determinar todas las estimaciones necesarias, se definirá en primer lugar, la metodología a usar en el proyecto, especificando sus fases y sus respectivas tareas. Seguidamente, se describirá el ciclo de vida del proyecto, y finalmente, se expondrá la planificación total.

3.2 Metodología del proyecto

Para este proyecto, la metodología adoptada es una métrica 3 con modificaciones para adecuar las fases a los requerimientos de nuestro proyecto pero con toda su esencia.

Esta métrica pretende alcanzar una serie de objetivos como proporcionar o definir sistemas de información que ayuden a conseguir los fines del cliente, mediante la definición de un marco estratégico de desarrollos de los mismos.

Dotar a los clientes, de productos software que satisfagan los requerimientos de los usuarios, dando un especial énfasis al análisis de requisitos. De éste modo, se tendrá la certeza de que el diseño y la implementación son lo suficientemente robustas.

Facilitar la comunicación entre los distintos participantes de la producción del software. El esquema a seguir según la metodología 3 adoptada por el equipo de desarrollo, es la siguiente:

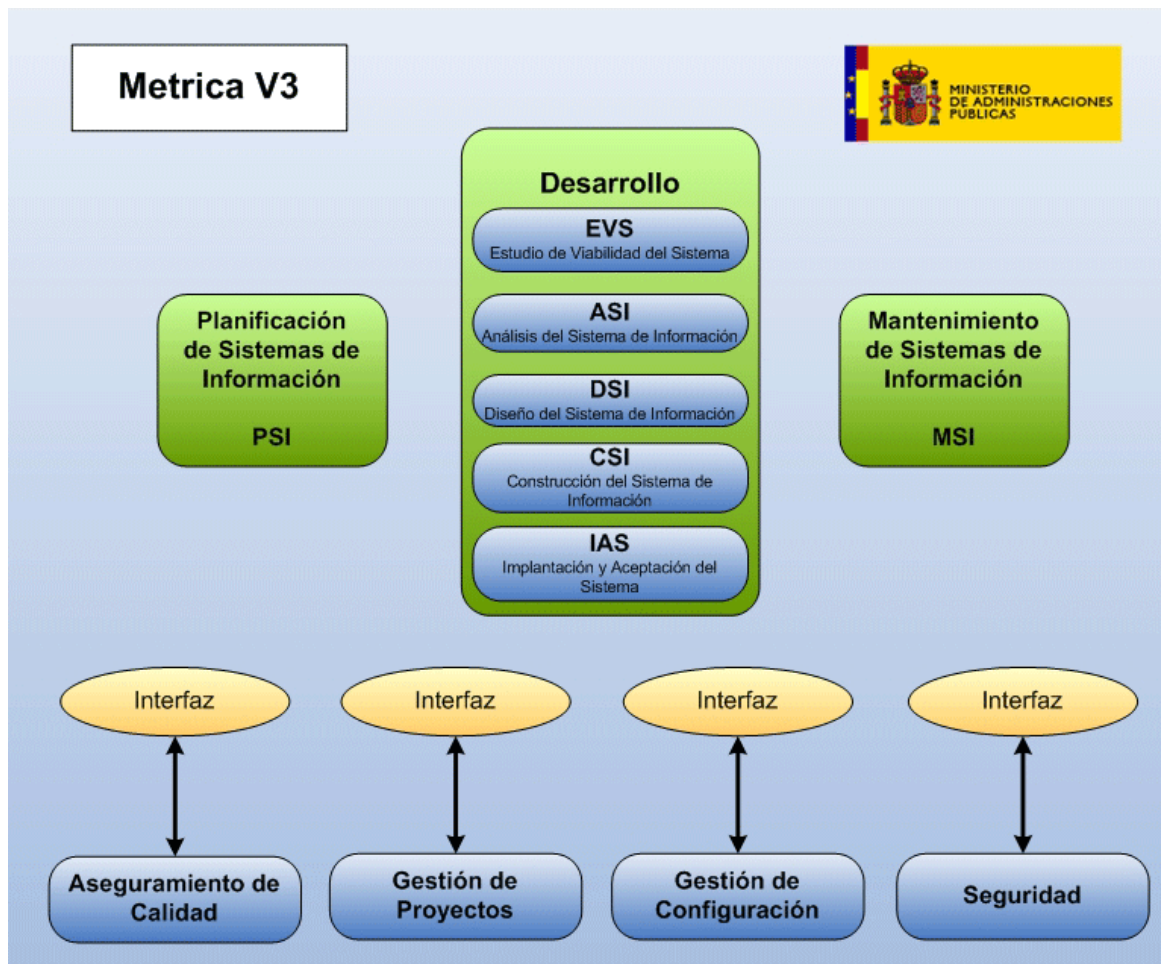


Figura 8. Modelo de calidad Metrica 3

3.2.1 Fases de la metodología adoptada

Análisis del Sistema de Información (ASI):

El objetivo de esta fase es la de recoger de forma detallada los requisitos de usuario y de software. De esta forma, se controla lo que el sistema de información debe cubrir.

Debe conseguir la especificación detallada del sistema de información, a través de un catálogo de requisitos y una serie de modelos que cubran las necesidades de información de los usuarios para los que se desarrollará el sistema de información y que serán la entrada para el proceso de Diseño del Sistema de Información (DSI).

Finalmente, decir, que debe obtener la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, además de los componentes del sistema de información.

Diseño del Sistema de Información (DSI):

El diseño de la arquitectura del sistema, dependerá de una gran variedad de las características del sistema. Es por ello que un buen trabajo en la fase ASI, simplificará mucho la tarea del DSI.

De este modo, tras esta tarea, se han de generar todas las especificaciones de construcción relativas al propio sistema, así como la especificación técnica del plan de pruebas, **la definición de requisitos de implantación** y el diseño de los procedimientos de migración y **carga inicial de datos**.

Para el presente proyecto, la fase de carga inicial de datos por parte del administrador del sistema, resultará de vital importancia. En este proceso se insertará información relativa a los usuarios de la aplicación.

Construcción del Sistema Informático (CSI):

A estas alturas del proyecto, para el inicio de esta fase, se toma como punto de partida los componentes diseñados previamente durante el proceso previo al de Diseño de Sistemas de Información (DSI).

Esta fase, debe cumplir con la construcción y prueba de los distintos componentes del sistema de información, de tal modo que a partir de un conjunto de especificaciones lógicas y físicas del mismo, se obtendrá el proceso de diseño del sistema de información.

Implantación y Aceptación del Sistema (IAS):

Para el inicio de este proceso se toman como punto de partida los componentes del sistema probados. De esta forma,

se deben haber probado de forma unitaria y se podrán integrar en el proceso de construcción del sistema de información (CSI).

Debe de realizar la entrega y aceptación del sistema en su totalidad y llevar a cabo las actividades oportunas para el paso a producción del sistema, en nuestro caso, al Market de Google. Además de la elaboración del plan de mantenimiento del sistema de forma que el responsable del mantenimiento lo conozca antes de que éste pase a producción.

Finalmente, decir que se ha desechado la fase de Mantenimiento del sistema definida en la métrica tres como fase final, Ello se ha debido a que éste equipo de desarrollo no iba a ser el encargado de llevarla a cabo, y se ha preferido delegar dicho trabajo a los que serán, sus futuros responsables. Además, comentar que toda la información de ésta métrica ha sido extraída de su documentación oficial.

3.3 Ciclo de vida

El ciclo de vida de un producto software como el que se va a desarrollar, define el orden para las tareas o actividades a acometer. Ciertos desarrollos software requieren un tiempo y esfuerzo determinado que no siempre será igual, por ello, el orden en las tareas que se describirán a continuación, pueden variar.

Las etapas principales a acometer en el ciclo de vida software, son:

- **Análisis:** Construye un modelo de requisitos y expone los casos de uso y posibles escenarios de la aplicación.
- **Diseño:** A partir del modelo de análisis se deducen las estructuras de datos, la estructura en la que se ha de descomponer el sistema y las interfaces necesarias a implementar.
- **Codificación:** Construye el sistema. La salida de esta fase concluye en un código ejecutable.
- **Pruebas:** Se comprueba que se cumplen ciertos criterios de corrección y calidad.
- **Implantación:** Esta fase comprende la entrega del software en la plataforma destinada para el usuario.
- **Mantenimiento:** En esta fase, que tiene lugar después de la entrega del software al cliente se asegura que el sistema siga funcionando y adaptándose a nuevos requisitos.

A continuación, se muestra una ilustración en la que se pretende explicar las fases y el orden de las que consta el ciclo de vida de este proyecto:

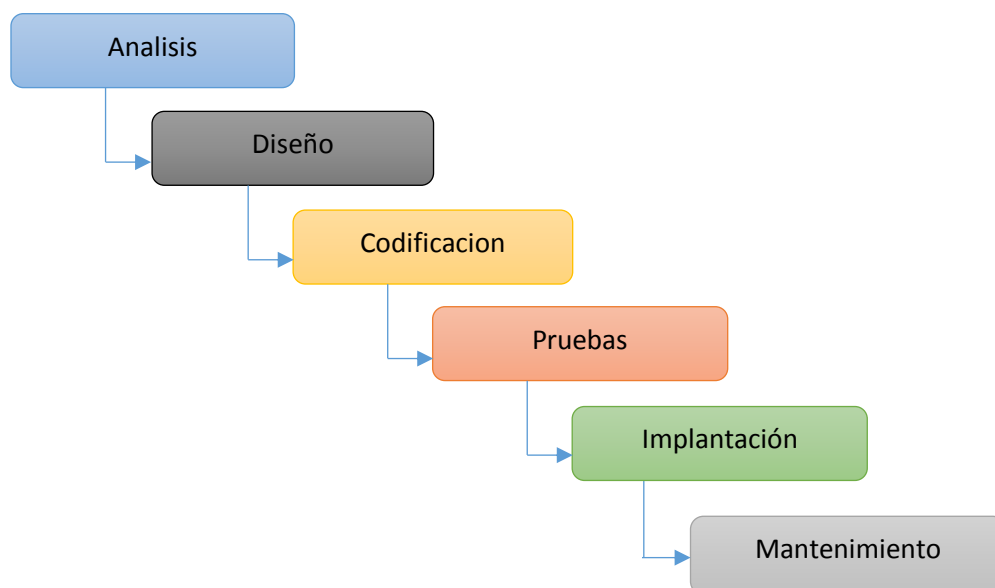


Figura 9. Ciclo de vida

3.4 Planificación

En este apartado se especificará la planificación de cada una de las tareas de las tres iteraciones en las que se dividirá el proyecto. Como se puede observar, las tareas tienen periodos distintos para cada iteración. Ello se debe a que en función de las necesidades del proyecto se estima que las mejoras impliquen un periodo específico en cada fase. Las tareas de diseño son las más extensas de forma general.

TAREA	FECHA INICIO	DURACION	FECHA FIN
Estudio propuesta inicial	03/01/2016	2	05/01/2016
Documentacion	05/01/2016	16	21/01/2016
Analisis	21/01/2016	28	18/02/2016
Diseño	18/02/2016	32	21/03/2016
Implementacion	21/03/2016	112	11/07/2016
Pruebas	11/07/2016	20	31/07/2016
Documentacion final	31/07/2016	53	27/09/2016

Tabla 3. Detalle tareas

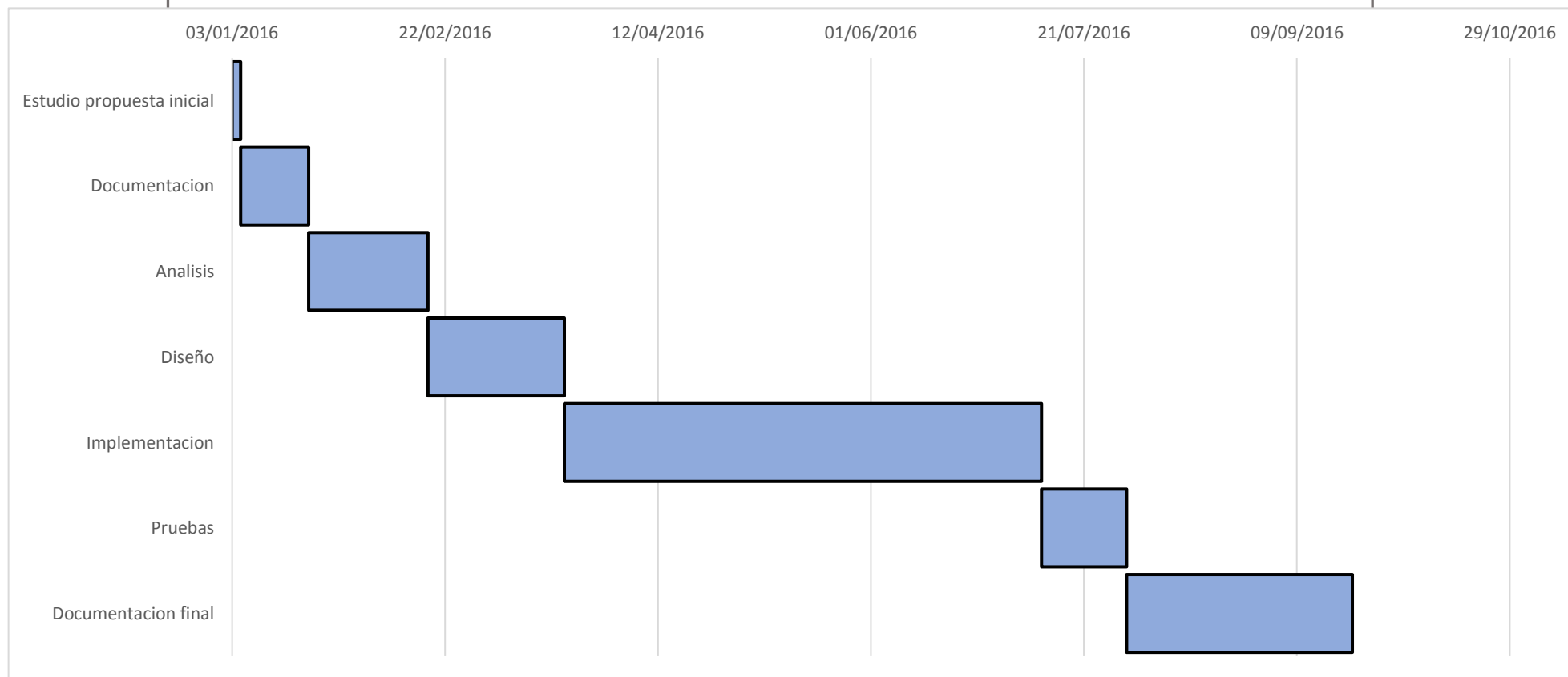


Figura 10. Diagrama de Gantt

3.5 Presupuesto

En esta sección se presenta el presupuesto estimado para el proyecto desarrollado, así como los costes asociados al mismo.

Todos los costes se calculan sin I.V.A., este se aplica en el resumen de costes totales del proyecto que se muestra tras el cálculo detallado de cada parte.

3.5.1 Resumen de tiempo dedicado

El diagrama de Gantt desarrollado en el apartado anterior posibilita el cálculo del número de horas totales dedicadas al proyecto.

A continuación se muestra en detalle el cálculo de horas dedicadas por separado a cada tarea, y la suma en conjunto a la realización de todas ellas.

- Propuesta inicial: 2 días x 8 horas = 16 horas.
- Documentación: 16 días x 8 horas = 128 horas.
- Análisis: 28 días x 8 horas = 224 horas.
- Diseño: 32 días x 8 horas = 256 horas.
- Implementación: 112 días x 8 horas = 896 horas.
- Pruebas: 20 días x 8 horas = 160 horas.
- Documentación final: 45 días x 8 horas = 360 horas.

Haciendo recuento del total de días y horas dedicadas al proyecto asciende a 263 días, lo que equivale a un total de 2040 horas.

3.5.2 Desglose: Coste de personal

A lo largo de toda la duración del proyecto únicamente ha desarrollado su trabajo un solo ingeniero junior. El total de días trabajados ha ascendido a 263 días, considerando una estimación de 8 horas de trabajo por día laborable. Se han tenido en cuenta como días libres de trabajo los festivos nacionales, locales y autonómicos.

Se ha de calcular el coste de personal dependiendo del rol desempeñado y el tiempo invertido por cada uno de ellos en el desarrollo del proyecto [54] [55]. A continuación se muestra el desglose para cada tipo de rol.

Apellidos y Nombre	Categoría	Dedicación (hombres mes) **	Coste hombre/mes (€)	Coste (€)
Santos Senra, Héctor	Analista	1	3.289,54 €	9.868,62 €
Santos Senra, Héctor	Diseñador	1	2.289,54 €	2.289,54 €
Santos Senra, Héctor	Programador	1	1.694,39 €	8.471,95 €
TOTAL PERSONAL				20.630,11 €

Tabla 4. Desglose del coste de personal

** 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

Donde el coste ha sido calculado a partir de la siguiente ecuación.

$$\text{Coste personal} = \frac{\text{Total días} * \text{horas/día}}{\text{Dedicación hombre/mes}} * \text{Coste hombre/mes}$$

3.5.3 Desglose: Coste de hardware

En los costes de hardware se incluyen todos los elementos hardware que han sido necesarios para el desarrollo del proyecto. A continuación se detallan los datos de ambos dispositivos, para el cual se ha calculado el coste imputable a partir de la fórmula de amortización siguiente:

$$\frac{A}{B} * C * D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto.

Concepto	Coste	% Uso dedicado al proyecto	Dedicacion (meses)	Periodo de deprecación	Coste imputable (€)
Ordenador portátil Portátil Dell XPS 15" i5 3.0 GHz 6GB RAM 500GB Disco duro.	1100 €	100	9	60	165
Raspberry Pi 3 MODELO B+ 1 GB RAM - WIFI - BLUETOOTH (BLE)	45.83€	100	9	60	6,87
Sensor Temperatura/Humedad DTH-11	13,50 €	100	9	60	2,02
Sensor movimiento PIR	14,70 €	100	9	60	2,20
Sensor LDR	2,85 €	100	9	60	0,42
Motor DC 6V - 100:1	13.18€	100	9	60	1,97
Fuente alimentación conmutada formato caja 25W - 5V	14.13€	100	9	60	2,11
TOTAL	180,61 €				

Tabla 5. Desglose del coste de Hardware

Como se resume en el total de la tabla anterior, el coste imputable correspondiente al hardware adquirido asciende a la cantidad de 180,61,8 €.

3.5.4 Desglose: Coste de software

A pesar de que la mayor parte del software con el que se ha trabajado es de código libre, y por tanto no tienen coste alguno, ha sido necesaria la adquisición de diversas licencias software para el desarrollo de ciertas fases del proyecto.

A continuación, se detallan las licencias y el coste sin I.V.A. asociado.

Concepto	Precio/unidad	Cantidad	Total
Microsoft Office Professional 2007	499 €	1	1100,00 €
Microsoft Windows 8 Pro	399€	1	45.83 €
TOTAL SOFTWARE			1291,00 €

Tabla 6. Desglose del coste de Software

3.5.5 Resumen de costes

Tras analizar los distintos costes asociados al proyecto, se resume a continuación cada uno de ellos y se establece el coste total del proyecto. A la suma de costes calculada, se añade un 20% de costes indirectos para cubrir los riesgos del proyecto y los gastos que no han sido tenidos en cuenta al realizar el presupuesto.

Se detalla el coste total con y sin I.V.A.

Concepto	Total
Costes de personal	65.280,00 €
Costes de amortización	180,61 €
Costes de software	1.291,00 €
Costes indirectos (20%)	13.350,32 €
Total sin I.V.A	80.101,93 €
Total con I.V.A (21%)	96.923,33 €

Tabla 7. Desglose del coste total

4. Análisis del sistema

En este apartado, se realizará un análisis en profundidad del sistema a desarrollar. Para ello, se comenzará por una introducción al sistema y seguidamente se describirán los requisitos del sistema. Éstos, se dividen principalmente en dos, los requisitos de usuario y de software.

Además, para finalizar el apartado, se expondrán los casos de uso del sistema. Definiendo todos los posibles escenarios de uso tanto de forma gráfica como textual.

4.1 Introducción

Como ya se ha detallado anteriormente se trata de implementar y desarrollar un sistema domótico que permita la monitorización de ciertos sensores, así como la manipulación de elementos finales en un modelo físico de vivienda.

4.2 Casos de Uso

Este apartado recoge el conjunto de situaciones en las que puede encontrarse el usuario mientras utiliza la aplicación. Para reflejarlo, utilizaremos un actor que representará al usuario y veremos, a través de un diagrama, cómo interactúa con las distintas actividades.

Para facilitar la lectura del diagrama, organizaremos la información en tablas cuyo formato incluirá los siguientes campos:

- Nombre: Nombre identificativo del caso de uso. Ha de ser unívoco y se utilizará a modo de resumen del mismo. No tiene que incluir detalles, pero sí ser descriptivo.
 - ID: identificador unívoco del requisito de usuario. Su nomenclatura ha de seguir el formato: CU-XX, donde:
 - XX será sustituido por el número de caso de uso. Comenzará en 01 e incrementará en una unidad a nuevo caso de uso.
 - Actor: Agente externo que interactúa con el sistema.
 - Objetivo: Acción que se persigue tras la interacción del actor con el sistema.
 - Pre-condición: Condición que debe cumplirse a priori para hacer el caso de uso.
 - Escenario: Descripción detallada de los pasos que ha de realizar el actor dentro del escenario para completar el caso de uso.
 - Escenario Alternativo: Descripción detallada de los pasos que ha de realizar el actor dentro del escenario, en caso de que pueda darse una bifurcación en la ejecución normal del sistema.
 - Post-condición: Hecho que se cumple si el flujo de eventos del Escenario se ejecuta correctamente.
-

Nombre	Descripción del caso de uso
Actores	
Descripción	
Pre-condición	
Escenario	
Escenario alternativo	
Post-condición	

Tabla 8. Detalle Caso de uso

CU-01	Identificarse en la aplicación
Actores	Usuario
Descripción	Permitir al usuario iniciar sesión en la aplicación
Pre-condiciones	La aplicación debe ser visible desde el navegador web.
Escenario	<ol style="list-style-type: none"> 1. El usuario introduce en el formulario de inicio de sesión los datos solicitados, correo electrónico y contraseña. 2. El usuario pulsa el botón Iniciar Sesión
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condiciones	El usuario visualiza la pantalla principal de la aplicación.

Tabla 9. Caso de uso 01

CU-02	Monitorización ambiental
Actores	Usuario
Descripción	Visualizar los valores ambientales de temperatura y humedad
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe estar identificado en la aplicación 2. El usuario debe encontrarse en la interfaz de Ambiente
Escenario	<ol style="list-style-type: none"> 1. El usuario se identifica correctamente en la aplicación 2. Se navega por la interfaz principal para seleccionar la opción Ambiente. 3. Se muestra la interfaz de monitorización ambiental para las distintas estancias 4. La interfaz muestra el dato correcto.
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condiciones	El usuario visualiza los datos ambientales que desea.

Tabla 10. Caso de uso 02

CU-03	Monitorización de la iluminación
Actores	Usuario
Descripción	Visualizar los valores de iluminación
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe estar identificado en la aplicación 2. El usuario debe encontrarse en la interfaz de Iluminación
Escenario	<ol style="list-style-type: none"> 1. El usuario se identifica correctamente en la aplicación 2. Se navega por la interfaz principal para seleccionar la opción Iluminación. 3. Se muestra la interfaz de iluminación para las distintas estancias 4. La interfaz muestra el dato correcto.
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condición	El usuario visualiza los datos de iluminación que desea.

Tabla 11. Caso de uso 03

CU-04	Modificación estado leds
Actores	Usuario
Descripción	Modificar el estado actual de un led
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe estar identificado en la aplicación 2. El usuario debe encontrarse en la interfaz de Iluminación
Escenario	<ol style="list-style-type: none"> 1. El usuario se identifica correctamente en la aplicación 2. Se navega por la interfaz principal para seleccionar la opción Iluminación. 3. Se muestra la interfaz de iluminación para las distintas estancias 4. El usuario selecciona la acción que desea llevar a cabo 5. La interfaz se actualiza mostrando el nuevo valor actual del led.
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condición	El usuario realiza correctamente el encendido o apagado del led.

Tabla 12. Caso de uso 04

CU-05	Monitorización estado leds
Actores	Usuario
Descripción	Consultar el estado actual de un determinado led
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe estar identificado en la aplicación 2. El usuario debe encontrarse en la interfaz de Iluminación
Escenario	<ol style="list-style-type: none"> 1. El usuario se identifica correctamente en la aplicación 2. Se navega por la interfaz principal para seleccionar la opción Iluminación. 3. Se muestra la interfaz de iluminación para las distintas estancias 4. El usuario comprueba el estado del led deseado.
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condición	El usuario visualiza el estado actual del led deseado

Tabla 13. Caso de uso 05

CU-06	Modificación estado motores
Actores	Usuario
Descripción	Cambiar el estado de un motor determinado
Pre-condiciones	El usuario debe estar identificado en la aplicación
Escenario	<ol style="list-style-type: none"> 1. El usuario se identifica correctamente en la aplicación 2. Se navega por la interfaz principal para seleccionar la opción Iluminación. 3. Se muestra la interfaz de iluminación para las distintas estancias 4. El usuario selecciona la acción que desea llevar a cabo 5. La interfaz se actualiza mostrando el nuevo valor actual del led
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condición	El usuario realiza correctamente la acción de modificar estado de motor

Tabla 14. Caso de uso 06

CU-07	Visualización de historial
Actores	Usuario
Descripción	Consultar historial de variables físicas
Pre-condiciones	El usuario debe estar identificado en la aplicación
Escenario	<ol style="list-style-type: none"> 1. El usuario se identifica correctamente en la aplicación 2. Se navega por la interfaz principal para seleccionar la opción Iluminación. 3. Se muestra la interfaz de iluminación para las distintas estancias 4. El usuario selecciona la acción que desea llevar a cabo 5. La interfaz se actualiza mostrando el nuevo valor actual del led.
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condición	El usuario visualiza el historial para cada una de las variables.

Tabla 15. Caso de uso 07

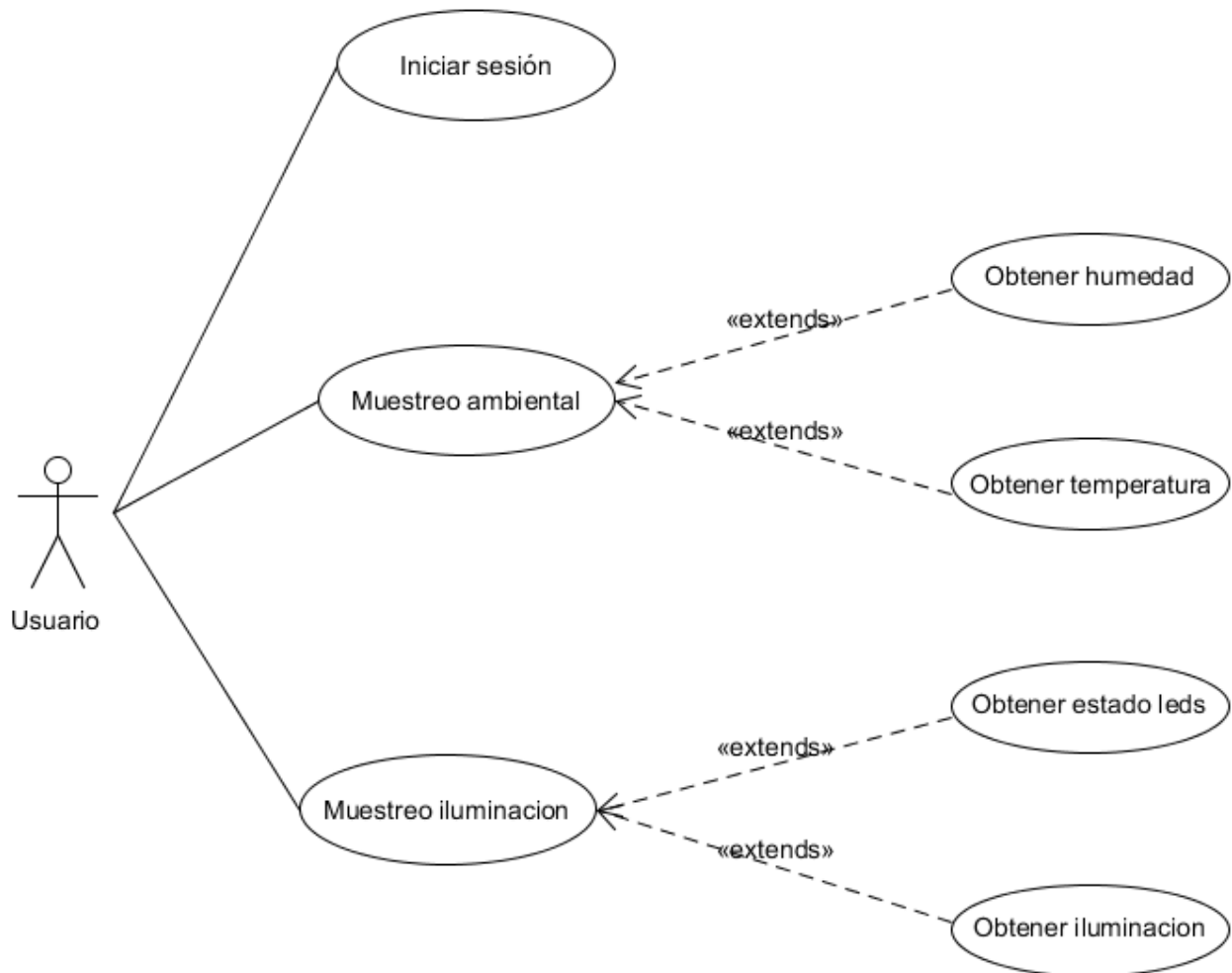
CU-08	Visualización de mapa
Actores	Usuario
Descripción	Consultar todos los datos de monitorización en todas las estancias.
Pre-condiciones	El usuario debe estar identificado en la aplicación
Escenario	<ol style="list-style-type: none"> 1. El usuario se identifica correctamente en la aplicación 2. Se navega por la interfaz principal para seleccionar la opción Iluminación. 3. Se muestra la interfaz de iluminación para las distintas estancias 4. El usuario selecciona la acción que desea llevar a cabo 5. La interfaz se actualiza mostrando el nuevo valor actual del led.
Escenario alternativo	El sistema informa al usuario si se producen errores
Post-condición	El usuario visualiza el mapa completo de todos los elementos.

Tabla 16. Caso de uso 08

CU-09	Creación de usuario por administrador
Actores	Administrador
Descripción	El administrador debe crear un usuario y contraseña para permitir el acceso a la aplicación
Pre-condiciones	El administrador debe tener acceso al servidor de base de datos
Escenario	<ol style="list-style-type: none"> 1. El usuario introduce en el formulario de inicio de sesión los datos solicitados, correo electrónico y contraseña. 2. El usuario pulsa el botón Iniciar Sesión
Escenario alternativo	El sistema informa al administrador si se producen errores
Post-condición	El administrador crea correctamente las credenciales de acceso

Tabla 17 Caso de uso 09

4.3 Diagramas UML

*Figura 11. Casos de uso CU-1, CU-2 Y CU-3*

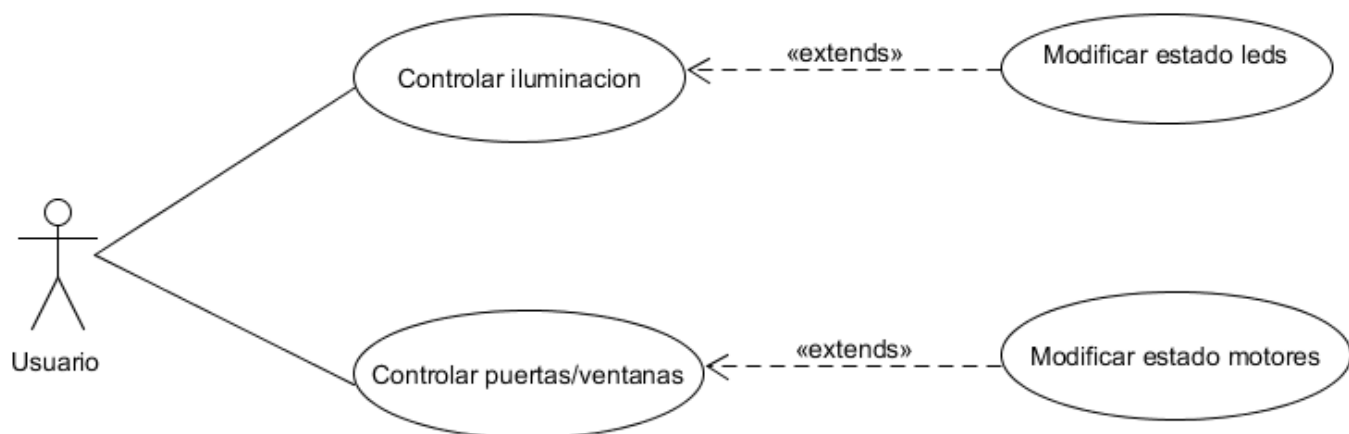


Figura 13. Casos de uso CU-4 y CU-6

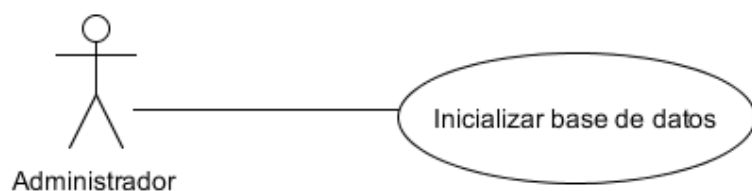


Figura 12. Casos de uso CU-7 y CU-8

4.4 Requisitos del Sistema

En primer lugar, se procederá a la extracción de requisitos del sistema. Éstos, se deberán diferenciar en dos tipos. En primer lugar, se definirán los requisitos de usuario, y seguidamente, los del software. Debido a que cada uno de ellos tiene un propósito se va a tratar de definir brevemente el de cada uno de ellos.

El requisito de usuario, pueden ser una descripción de lo que un sistema debe hacer. Este tipo de requisito, especifica lo que el sistema entregado debe ser capaz de realiza.

El requisito de software por otro lado, pretende especificar algo sobre el propio sistema y cómo ha de realizar las funciones deseadas.

4.4.1 Requisitos de Usuario

- **Identificador:** Identifica de forma unívoca cada uno de los requisitos de usuario que se explicarán a continuación. El formato para los requisitos de usuario de capacidad será RUC-XX, siendo XX la numeración del requisito. Del mismo modo, el formato para los requisitos de usuario de restricción, será RUR-XX.
- **Prioridad:** Indica la prioridad de implementación del requisito según el impacto en el desarrollo del mismo. Sus posibles valores, serán: “alta”, “media” o “baja”.
- **Fuente:** Definirá el origen del requisito, de donde surge la extracción del mismo. Sus valores posibles, será “cliente” o “expertos”.
- **Necesidad:** Indicará la necesidad de implementación del requisito. Sus posibles valores, serán “esencial”, “deseable” y “opcional”.
- **Estabilidad:** Dado que algunos requisitos pueden permanecer estables durante el ciclo completo de vida del software, se deben distinguir aquellos susceptibles de ser cambiados en un futuro próximo. En este caso, los posibles valores para éste atributo, será “estable” en caso de que no se prevea su cambio y “inestable” si estima que en un futuro podrá ser modificado cierto aspecto del sistema.
- **Descripción:** Expone brevemente el requisito.

4.4.1.1 Requisitos de capacidad

Este apartado recoge los requisitos de usuario, que se dividirán en dos categorías: requisitos de usuario de capacidad y requisitos de usuario de restricción

1. Identificador		RUC/RUR - XX	
Prioridad	Alta/Media/Baja	Fuente	Cliente/Experto
Necesidad	Esencial/Deseable/Opcional	Estabilidad	Estable/Inestable
Descripción	Descripción del requisito de usuario		

Tabla 18. Detalle Requisitos de usuario

Identificador		RUC - 01	
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	Contar con un dispositivo que permita visualizar la aplicación web.		

Tabla 19. Requisito de capacidad visualizar aplicación

Identificador		RUC - 02	
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario debe ser capaz de identificarse en la aplicación.		

Tabla 20. Requisito de capacidad identificar usuario

Identificador		RUC - 03	
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario deberá ser capaz de visualizar el valor instantáneo de la información recogida por los sensores.		

Tabla 21. Requisito de capacidad leer valor instantáneo

Identificador		RUC - 04	
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario deberá ser capaz de visualizar el valor medio de la información recogida por los sensores.		

Tabla 22. Requisito de capacidad leer valor medio

Identificador		RUC - 05	
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario deberá de ser capaz de consultar un histórico diario de las variables que permite monitorizar el sistema: luminosidad, presencia, temperatura y humedad.		

Tabla 23. Requisito de capacidad consultar histórico

Identificador		RUC - 06	
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario deberá de ser capaz de controlar leds y motores (actuadores) habilitados para ello en el modelo físico.		

Tabla 24. Requisito de capacidad controlar actuadores

Identificador		RUC - 07	
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario deberá de ser capaz de visualizar el estado de actuadores en un momento dado.		

Tabla 25. Requisito de capacidad visualizar estado actuadores

4.4.1.2 Requisitos de restricción

Identificador	RUR - 01		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario no podrá acceder a la aplicación web a través de internet.		

Tabla 26. Requisito de restricción limitar uso internet

Identificador	RUR - 02		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario no podrá consultar información para días distintos al actual.		

Tabla 27. Requisito de restricción limitar plazo consulta

4.4.2 Requisitos del Software

Los requisitos de software funcionales, pretenden definir el comportamiento interno del software, detalles técnicos, manipulación de datos y otras funcionalidades del sistema.

Los requisitos de software no funcionales, se enfocan al diseño o la implementación del sistema.

El formato de la tabla de requisitos de software tanto funcional como no funcional, se mantiene idéntica a la anterior. Sin embargo, los identificadores cambian de formato, de tal modo que quedan como:

- RSF – XX: Representa el identificador para los Requisitos de Software Funcionales. Además el campo XX identifica el valor numérico del requisito, el cual será único.
- RSNF – XX: Representa el identificador para los Requisitos de Software No Funcionales. Además el campo XX identifica el valor numérico del requisito, el cual será único.

Finalmente, decir que para éste tipo de requisitos la fuente será un Requisito de Usuario de, tanto de Capacidad como de Restricción (RUR o RUC). Además, cabe añadir que varios Requisitos de Software (RSF o RSNF) pueden corresponder con un único Requisito de Usuario (RUR o RUC)

1. Identificador		RUC/RUR - XX	
Prioridad	Alta/Media/Baja	Fuente	Cliente/Experto
Necesidad	Esencial/Deseable/Opcional	Estabilidad	Estable/Inestable
Descripción	Descripción del requisito de usuario		
Fuente	RUC-XX/RUC-YY		

Tabla 28. Detalle Requisito de software

4.4.2.1 Requisitos funcionales

Identificador		RSF - 01	
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	Contar con un dispositivo telemático con navegador web.		
Fuente	RUC-01		

Tabla 29. Requisito funcional dispositivo telemático

Identificador		RSF - 02	
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	El dispositivo telemático deberá tener conexión a la red donde se encuentra instalado el sistema domótico.		
Fuente	RUC-01		

Tabla 30. Requisito funcional dispositivo con conexión

Identificador		RSF - 03	
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema debe ser capaz de crear nuevos usuarios para la aplicación.		
Fuente	RUC-02		

Tabla 31. Requisito funcional crear usuarios

Identificador		RSF - 04	
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema debe ser capaz de identificar cualquier acceso realizado a la aplicación.		
Fuente	RUC-02		

Tabla 32. Requisito funcional identificar accesos

Identificador		RSF - 05	
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema debe ser capaz de controlar accesos no autorizados a la aplicación.		
Fuente	RUC-02		

Tabla 33. Requisito funcional control accesos

Identificador		RSF - 06	
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de leer el dato recogido por cualquier sensor en un momento dado.		
Fuente	RUC-03		

Tabla 34. Requisito funcional leer dato sensor

Identificador		RSF - 07	
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de mostrar en una interfaz la información instantánea recogida por los sensores.		
Fuente	RUC-03		

Tabla 35. Requisito funcional mostrar dato

Identificador	RSF - 08		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de calcular el valor medio de temperatura recogido por los sensores.		
Fuente	RUC-04		

Tabla 36. Requisito funcional calculo valor medio

Identificador	RSF - 09		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de almacenar el valor medio de temperatura recogido por los sensores.		
Fuente	RUC-04		

Tabla 37. Requisito funcional almacenar valor medio

Identificador	RSF - 10		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de almacenar en la base de datos la información recogida por los sensores.		
Fuente	RUC-05		

Tabla 38. Requisito funcional almacenar valores

Identificador	RSF - 11		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de mostrar en una interfaz la información almacenada relativa a los sensores, a lo largo de un día.		
Fuente	RUC-05		

Tabla 39. Requisito funcional mostrar historico

Identificador	RSF - 12		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de ejecutar consultas sobre el servidor de base de datos de la aplicación.		
Fuente	RUC-05		

Tabla 40. Requisito funcional realizar consultas

Identificador	RSF - 13		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá de ser capaz de modificar el estado de los actuadores.		
Fuente	RUC-06		

Tabla 41. Requisito funcional modificar actuadores

Identificador	RSF - 14		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema permitirá consultar el estado de los actuadores.		
Fuente	RUC-07		

Tabla 42. Requisito funcional consultar estado actuadores

Identificador	RSF - 15		
Prioridad	Alta	Fuente	Experto
Necesidad	Esencial	Estabilidad	Estable
Descripción	El sistema deberá almacenar en la base de datos las acciones realizadas sobre los actuadores.		
Fuente	RUC-07		

Tabla 43. Requisito funcional almacenar acciones

4.4.2.2 Requisitos no funcionales

Identificador	RSNF - 01		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario no podrá acceder a la aplicación web a través de internet.		

Tabla 44. Requisito restricción conexión internet

Identificador	RSNF - 03		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario no podrá consultar información para días distintos al actual.		

Tabla 45. Requisito restricción plazo consulta

Identificador	RSNF - 02		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario no podrá crear nuevos perfiles para la aplicación.		

Tabla 46. Requisito restricción crear usuarios

Identificador	RSNF - 04		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Descripción	El usuario no podrá consultar valores medios de luminosidad.		

Tabla 47. Requisito restricción valor medio luminosidad

4.5 Matriz de trazabilidad

Una vez recogidos y analizados tanto los requisitos de usuario como los requisitos software, ha de comprobarse que todo requisito de usuario esté contemplado por al menos un requisito software.

Para hacer esta comprobación, se utilizan matrices de trazabilidad. Una matriz de trazabilidad tiene en su eje vertical el conjunto de identificadores de requisitos de usuario, tanto de capacidad como de restricción, y en su eje horizontal, los identificadores de requisitos software.

Si recordamos del apartado anterior, los requisitos software tienen un campo llamado Dependencias UR que indica el requisito de usuario evaluado o implementado por el requisito software en cuestión. Siendo así, la matriz de trazabilidad marcará cada requisito de usuario contemplado por un requisito software concreto, siempre y cuando este haya sido asignado en el campo Dependencias UR correspondiente.

Si una línea de la matriz quedara completamente en blanco, significaría que en la aplicación final no se vería implementado el requisito correspondiente a dicha fila y el proyecto quedaría incompleto. Para comprobar que esto no ocurre, analizamos las matrices de trazabilidad para las dos aplicaciones.

	RSF-01	RSF-02	RSF-03	RSF-04	RSF-05	RSF-06	RSF-07	RSF-08	RSF-09	RSF-10	RSF-11	RSF-12	RSF-13	RSF-14	RSF-15	
RUC-01	x	x														
RUC-02			x	x	x											
RUC-03						x	x	x								
RUC-04								x	x							
RUC-05									x	x	x					
RUC-06												x	x			
RUC-07														x	x	

Tabla 48. Matriz de trazabilidad. Requisitos de capacidad con requisitos funcionales

5. Descripción del sistema domótico

5.1 Introducción

En primer lugar se realizará una descripción de los elementos del sistema, utilizados para la recolección, monitorización, almacenamiento y posterior visualización. Se mostrará la distribución de elementos hardware instalados.

En segundo lugar se analizará el hardware necesario para el desarrollo del sistema como son los sensores, actuadores y el microcontrolador.

En tercer lugar, se expondrá la estructura del sistema y la interrelación existente entre los distintos elementos del sistema.

5.2 Requisitos del sistema domótico

Una de las características de cualquier sistema domótico actual es la capacidad de monitorizar las estancias del inmueble para conocer distintos factores como la temperatura y la humedad, la cantidad de luz, existencia de movimiento en la sala o cantidad de llamadas recibidas en la extensión correspondiente a la sala.

Para este proyecto se van a elegir una serie de variables físicas y ambientales de interés que serán medidas y recolectadas por sensores digitales instalados en cada una de las estancias.

Tipos de variables a monitorizar

Cada tipo de sensor permitirá registrar cada una de las siguientes variables físicas:

1. Variable temperatura/humedad
2. Variable presencia
3. Variable luminosidad

Para cada una de estas variables, se permitirá consultar un análisis histórico temporal mediante gráficas y clasificado en función de la estancia donde se desea conocer el estado de la variable en cuestión.

La temperatura y la humedad son variables de tipo continuo ya que los posibles valores que pueden tomar son infinitos dentro de unos puntos en sus correspondientes escalas.

INDICADOR	TEMPERATURA	NIVEL
TEMPERATURA ALTA	[28, 50] °C	2
TEMPERATURA MEDIA	[21, 27] °C	1
TEMPERATURA BAJA	[0, 20] °C	0

Tabla 49. Modelo temperatura



Figura 14. Icono temperatura

La **presencia** es una variable de tipo **discreto** ya que únicamente puede tomar los valores {0,1}. Para representar esta variable se ha escogido el logotipo de la figura



Figura 15. Icono presencia

PRESENCIA	
EXISTE MOVIMIENTO	1
NO EXISTE MOVIMIENTO	0

Tabla 50. Modelo presencia

Los niveles de apertura y cierre de la puerta del garaje, así como los de las ventanas se reflejarán mediante cinco valores, estableciendo el más bajo para una puerta o ventana completamente cerrada y el valor más alto para una apertura completa del elemento a controlar.



Figura 16. Icono seguridad

ESTADO VENTANAS/PUERTA	
VENTANA/PUERTA ABIERTA 100 %	4
VENTANA/PUERTA ABIERTA 75 %	3
VENTANA/PUERTA ABIERTA 50 %	2
VENTANA/PUERTA ABIERTA 25 %	1
VENTANA/PUERTA ABIERTA 0 %	0

Tabla 51. Modelo seguridad

Respecto a la iluminación se realizara un monitoreo básico de la misma, utilizando tres niveles distintos de luminosidad. La correspondencia entre los posibles valores que arroja el sensor y los niveles elegidos vendrá dada por los valores mínimo y máximo de luminosidad obtenidos en el análisis preliminar.

Los valores obtenidos representan numéricamente los cambios en la intensidad de la luz



Figura 17. Icono iluminación

LUZ	
NIVEL LUZ ALTO	2
NIVEL LUZ MEDIO	1
NIVEL LUZ BAJO	0

Tabla 52. Modelo iluminación

Distribución de elementos

SALON

- El salón dispondrá de un motor para regular la apertura de la ventana
- También dispondrá de un sensor LDR para medir la cantidad de luz
- También dispondrá de un sensor de movimiento para encender un LED en el salón



Figura 18. Representación interfaz salón

HALL

- El hall dispondrá de un sensor de movimiento para encender un LED en el hall (permanece encendido 10 seg.)
- Tambien dispondrá de un sensor LDR para saber si el LED esta encendido o apagado (medir la cantidad de **LUZ INTERIOR**)



Figura 19. Representación interfaz hall

HABITACION

- La única habitación de la casa dispondrá de un **sensor de temperatura/humedad**.
- Tambien dispondrá de un motor para regular la apertura de la ventana (MANUAL)
- Tambien dispondrá de un LED REGULABLE en la habitación



Figura 20. Representación interfaz habitación

BAÑO

- El baño dispondrá de un sensor de **temperatura/humedad**.



Figura 21. Representación interfaz cuarto de baño

COCINA

- La cocina dispondrá de un sensor de movimiento para encender un LED en la cocina.

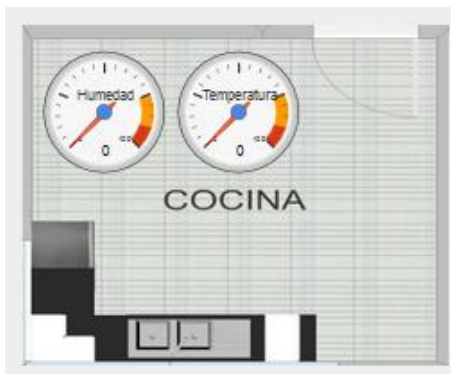


Figura 22. Representación interfaz cocina

GARAJE

- El garaje dispondrá de un sensor de movimiento para encender un LED en el garaje
- El garaje dispondrá de un motor para abrir o cerrar la puerta del garaje(MANUAL)



Figura 23. Representación interfaz garaje



Figura 24. Modelo físico vivienda

En la tabla XX se muestra una relacion de los elementos detallados anteriormente en una tabla con los identificadores correspondientes a cada elemento.

ELEMENTO	ESTANCIA	FUNCION
TEMP-1	BAÑO	sensor temperatura 1
TEMP-2	HABITACION	sensor temperatura 2
PIR-1	COCINA	sensor presencia 1
PIR-2	HALL	sensor presencia 2
PIR-3	GARAJE	sensor presencia 3
PIR-4	SALON	sensor presencia 4
LDR-1	HALL	sensor luz 1
LDR -2	SALON	sensor luz 2
LED-1	HALL	Led 1
LED-2	COCINA	Led 2
LED-3	HABITACION	Led 3
LED-4	GARAJE	Led 4
M-1	SALON	dc motor 1
M-2	HABITACION	dc motor 2
M-3	GARAJE	dc motor 3

Tabla 53. Detalle elementos modelo físico

5.3 Infraestructura del sistema

El sistema domótico que ocupa este proyecto implementa los elementos básicos de cualquier sistema de domótica habitual.

Un servidor domótico (cliente IP) que realizará el trabajo de enviar los mandatos a los dispositivos de su red. Además, será necesario un BUS de datos que sea capaz de interconectar el servidor domótico (ó central IP) con los dispositivos que se encuentren domotizados.

Es decir, deberá ser capaz de enviar de un punto a otro las peticiones de los dispositivos. Además, deberá contar con los dispositivos que automatizan mediante un 'actuador' las acciones a realizar.

Por ejemplo, el servidor domótico conectado a una red, recibirá una petición de tipo request HTTP, procedente de alguno de los clientes que se encuentran conectados a la aplicación. A partir de ese momento, el servidor, enviará dicha acción al dispositivo que corresponda, la acción a acometer por medio de un BUS de comunicación. Por último el dispositivo, deberá llevar a cabo la acción recibida.

A continuación, se enumeran los dispositivos necesarios:

- Servidor domotico
- Bus de comunicaciones
- Fuente de alimentación del sistema
- Sensores y actuadores

Servidor domotico

El micro-controlador elegido como servidor domotico ha sido el modelo Raspberry Pi 3, que cuenta con un procesador Broadcom BCM2387 con cuatro núcleos ARM Cortex-A53 a 1,2 GHz.

La tarea principal que desempeña el micro-controlador es la de ejecutar el sistema operativo Raspbian, que permitirá alojar tanto el servidor web (Apache), como el servidor de base de datos (MySQL). También se ocupa de enviar y recibir las ordenes necesarias para el correcto desempeño del sistema.

Bus de comunicaciones

El bus GPIO se situará entre el microcontrolador y los sensores/actuadores permitiendo la transmisión de información desde los sensores hacia el microcontrolador y en sentido contrario cuando se trate de controlar alguno de los actuadores.

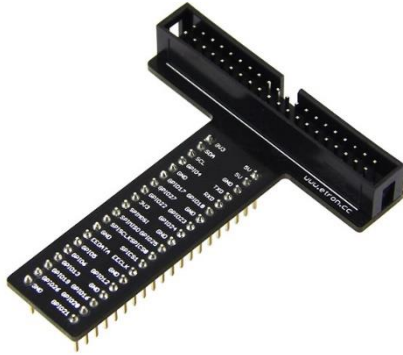


Figura 26. Conector bus GPIO

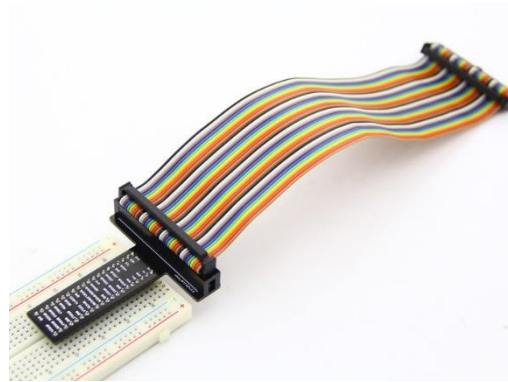


Figura 25. Bus GPIO

Fuente de alimentación del sistema

La fuente de alimentación será el elemento encargado de suministrar energía al conjunto de elementos electrónicos. Este modelo utilizado es de 25W 5V-5A con capacidad para poder alimentar tanto el microcontrolador como los motores.

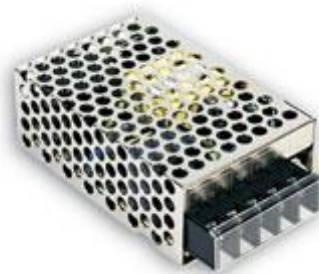


Figura 27. Fuente de alimentación

Sensores y actuadores

Sensores

Para este proyecto se han elegido una serie de sensores estándar de amplio uso en plataformas hardware como arduino y raspberry pi. Para cada variable a monitorizar se hará uso de sensores distintos, como son el DHT-11 para medir la temperatura y la humedad o el sensor infrarrojo pasivo (PIR) para detectar el movimiento.

Sensor Temperatura/Humedad DHT-11

El DHT11 es un sensor de temperatura y humedad digital de bajo costo. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica). Es bastante simple de usar, pero requiere sincronización cuidadosa para tomar datos. El único inconveniente de este sensor es que sólo se puede obtener nuevos datos una vez cada 2 segundos, así que las lecturas que se pueden realizar serán mínimo cada 2 segundos.

Características	
Alimentación	$3Vdc \leq Vcc \leq 5Vdc$
Rango de medición de temperatura	0 a 50 °C
Precisión de medición de temperatura	± 2.0 °C
Resolución Temperatura	0.1°C
Rango de medición de humedad	20% a 90% RH.
Precisión de medición de humedad	4% RH.
Resolución Humedad	1% RH
Tiempo de sensado	1 seg.

Tabla 54. Características sensor DHT-11



Figura 28. Sensor DHT-11

Movimiento

El módulo HC-SR501 tiene 3 pines de conexión +5v, OUT (3,3v) y GND, y dos resistencias variables de calibración (Ch1 y RL2).

Ch1: Con esta resistencia se permite establecer el tiempo que se va a mantener activa la salida del sensor. Una de las principales limitaciones de este módulo es que el tiempo mínimo que se puede establecer es de más o menos 3 segundos. Si cambiamos la resistencia por otra de 100K, podemos bajar el tiempo mínimo a más o menos 0,5 s.

RL2: Esta resistencia variable permite establecer la distancia de detección que puede variar entre 3-7m.



Figura 29. Sensor PIR

Características	
Sensor piroeléctrico (Pasivo) infrarrojo	
El módulo incluye el sensor, lente, controlador PIR BISS0001	
Rango de detección:	3 m a 7 m, ajustable mediante trimmer (Sx)
Lente fresnel de 19 zonas, ángulo < 100°	
Salida activa alta a 3.3 V	
Tiempo en estado activo de la salida configurable mediante trimmer (Tx)	
Redisparo configurable mediante jumper de soldadura	
Consumo de corriente en reposo	< 50 μ A
Voltaje de alimentación	4.5 VDC a 20 VDC

Tabla 55. Características sensor PIR

Luminosidad

El LDR (Light Dependent Resistor) o resistencia dependiente de la luz o también fotocélula, es una resistencia que varía su resistencia en función de la luz que incide sobre su superficie. Cuanto mayor sea la intensidad de la luz que incide en la superficie del LDR menor será su resistencia y cuanto menos luz incida mayor será su resistencia.

Se emplean en iluminación, apagado y encendido de alumbrado (interruptores crepusculares), en alarmas, en cámaras fotográficas, en medidores de luz. Las de la gama infrarroja en control de máquinas y procesos de conteo y detección de objetos.



Figura 30. Sensor LDR

Actuadores

Por otro lado, la capacidad de control sobre los elementos finales de un edificio o vivienda inteligente, es fundamental para asegurar algunos de los servicios que aporta un sistema domotico, como el confort, la accesibilidad o la seguridad.

Los elementos finales o actuadores que se pretenden manipular han sido elegidos por su fácil manejo y versatilidad. Se ha elegido por un lado un tipo de actuador booleano, como es un led cuyos estados serán encendido o apagado y por otro lado un motor de corriente continua que permite ser manipulado para controlar puertas y ventanas.

Se ha utilizado un pequeño motor de metal y de gran potencia con reductora para aplicaciones en robótica.

Características	
Dimensiones	24 x 10 x 12 mm
Ratio de la reductora	10:1
Diámetro del eje	3,9mm (con ranura de bloqueo)
Voltaje nominal	6Vcc (puede funcionar entre 3 a 9Vcc)
Velocidad de giro sin carga	3000rpm
Consumo sin carga	40mA (Max: 360mA)
Par	0,2 kg-cm (max)
Peso	10 gramos

Tabla 56. Características motor dc



Figura 31. Motor DC

Se puede observar en la Figura 1Figura 33 como los actuadores envían y reciben señales a través del bus mientras que los sensores únicamente introducen señales en el bus.

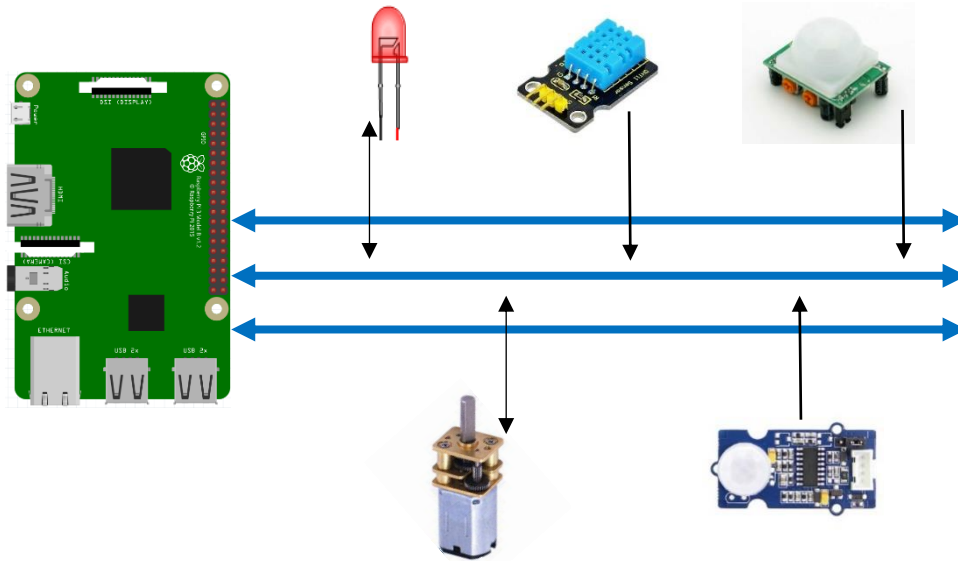


Figura 33. Diagrama sistema hardware

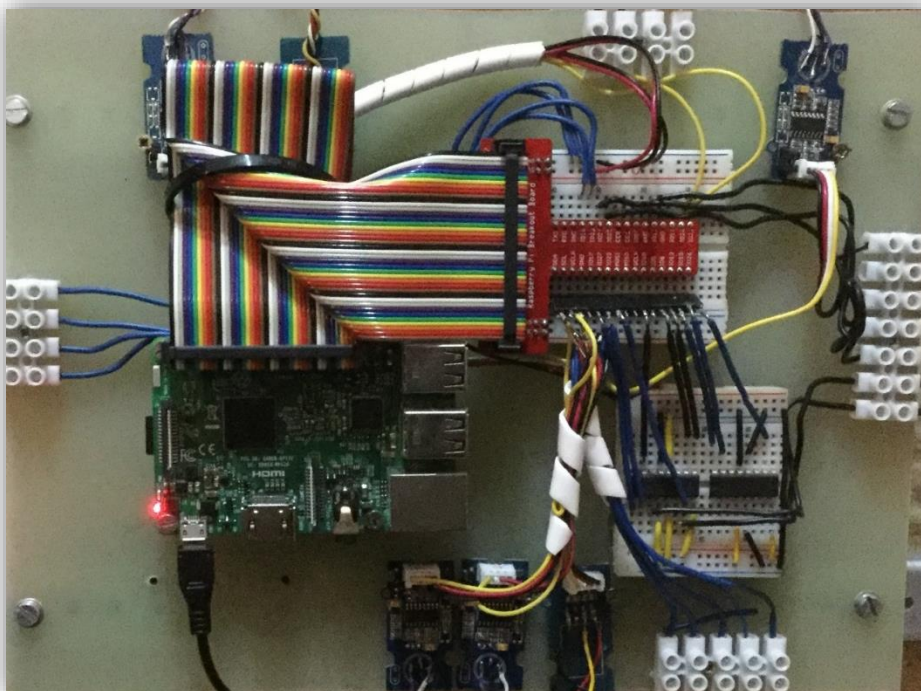


Figura 32. Montaje final modelo físico

6. Diseño del sistema

En este apartado se estudiarán los elementos relacionados con la fase de diseño del ciclo de desarrollo software. Se analizará el diseño arquitectónico del sistema, las interfaces con las que el usuario interactuará y la estructura de la base de datos con la que trabajará el servidor.

En primer lugar se expone el lenguaje de programación elegido para el desarrollo del sistema. En segundo lugar se detallará la estructura de la base de datos. Se terminará con el diseño de las comunicaciones entre los dos subsistemas y el diseño de las interfaces de usuario.

6.1 Lenguajes de programación

Como se ha comentado anteriormente, el sistema desarrollado se puede dividir a su vez en varios subsistemas como son la aplicación web junto al servidor web y por otro lado el servidor de base de datos junto a toda la infraestructura hardware requerida.

Para elegir el lenguaje de programación se ha tenido en cuenta principalmente el conocimiento y la experiencia del grupo de desarrollo con estos lenguajes. Una mayor familiaridad con los mismos dará como resultado unos tiempos menores de desarrollo y mejoras considerables en cuanto a la calidad del software se refiere.

6.1.1 El servidor web

En general el sistema se basa en una arquitectura Cliente-Servidor donde la aplicación cliente realizará peticiones al servidor. Este se encargará de dar respuesta a las peticiones del cliente, devolviendo la información requerida.

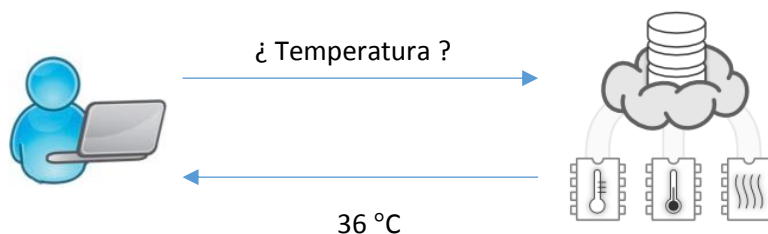


Figura 34. Ejemplo comunicación usuario servidor web

El servidor web será el encargado de almacenar la pagina web que utilizarán los usuarios para acceder a la aplicación. También deberá contener la lógica necesaria para procesar las peticiones realizadas por los usuarios a la aplicación.



Figura 35. Servidor LAMP

En el desarrollo de la página web se han utilizado lenguajes de diseño de interfaces como son HTML y CSS. También ha sido necesario el uso de Javascript para interactuar con las librerías gráficas de Google Charts.

HTML es un lenguaje de marcado que domina el campo del desarrollo de páginas web. Se utiliza para transformar unos determinados caracteres de texto en formas y figuras que dotan a la página de una mayor usabilidad.

CSS es un lenguaje utilizado para definir la presentación de un documento estructurado escrito en HTML o XML. EL W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores.

JavaScript es un lenguaje de programación interpretado. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas.

Para la gestión de las plantillas se ha hecho uso del framework de desarrollo Bootstrap, que facilita la labor de maquetación y permite una creación rápida y sencilla de interfaces atractivas.



Figura 36. Framework Bootstrap

6.1.2 Aplicación web

La aplicación web ha sido desarrollada en PHP. Desde este lenguaje se realizarán llamadas a scripts escritos en Python. Estos scripts accederán directamente al puerto GPIO del microcontrolador para acceder a la información proporcionado por los sensores y actuadores.

El micro controlador tiene predefinidas unas tareas de ejecución, definidas en el fichero crontab de Linux. Estas tareas se encargarán de llamar a los scripts correspondientes en cada momento.

Estos procesos también permitirán la obtención de datos automatizada desde los sensores, así como el correspondiente tratamiento de los datos.

Lenguaje PHP



Figura 37. Lenguaje PHP



Figura 38. Lenguaje Python

6.2 Arquitectura del Sistema

Actualmente, en el desarrollo de software es imprescindible la utilización de patrones de diseño que otorguen a las aplicaciones características que les permitan ser flexibles, evolutivas y de fácil mantenimiento.

El presente TFG se ha desarrollado utilizando el patrón Modelo-Vista-Controlador, a partir de ahora MVC, el cual se adapta perfectamente a la arquitectura Cliente-Servidor y permite una separación entre la lógica, los datos y la vista.

En el patrón MVC existen tres elementos:

Modelo: es el conjunto de clases que representa la información del mundo real que el sistema debe reflejar.

Vista: es la encargada de la representación visual del modelo al usuario

Controlador: recibe, trata y responde a los eventos enviados por el usuario. Interactúa tanto con el modelo como con la vista.

El flujo de control del MVC donde se detalla paso a paso como se realizarían las llamadas y comunicación en la aplicación es el siguiente:

1. Se realiza una petición desde la aplicación.
2. El controlador recibe la petición y la gestiona.
3. El controlador consulta o modifica el modelo.
4. El controlador envía la respuesta a la vista.
5. La vista reacciona a la respuesta modificando la interfaz.

Aunque originalmente MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón. Estos frameworks se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.

Los primeros frameworks MVC para desarrollo web planteaban un enfoque de cliente ligero en el que casi todas las funciones, tanto de la vista, el modelo y el controlador recaían en el servidor. En este enfoque, el cliente manda la petición de cualquier hipervínculo o formulario al controlador y después recibe de la vista una página completa y actualizada (u otro documento); tanto el modelo como el controlador (y buena parte de la vista) están completamente alojados en el servidor.

El modelo vista-controlador se representa en la Figura XXX, donde se puede apreciar el esquema de interacción entre el usuario y el sistema.

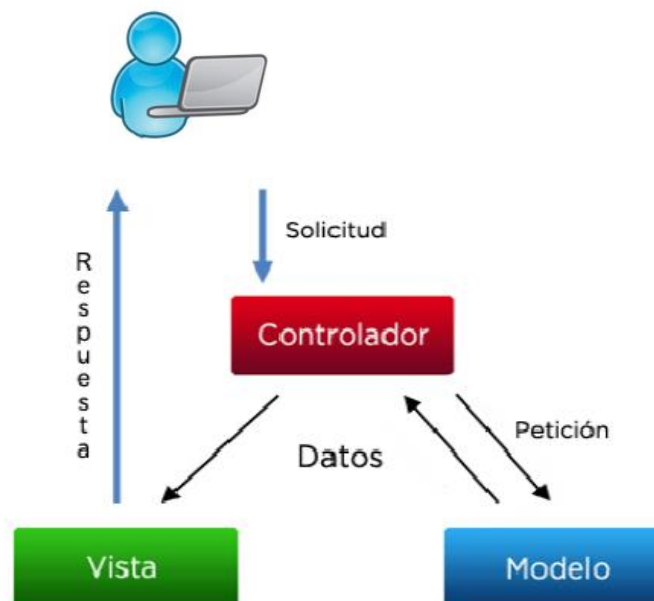


Figura 39. Esquema MVC

En la Figura 40 se puede apreciar el sistema en su conjunto. Por un lado el usuario realiza peticiones de tipo http post al servidor web contenido en el microcontrolador. Éste a su vez se encarga de recibir los datos provenientes de los sensores y de enviar las órdenes apropiadas a los actuadores.

En una etapa final el usuario recibe los datos solicitados o la confirmación de las acciones que ha realizado.

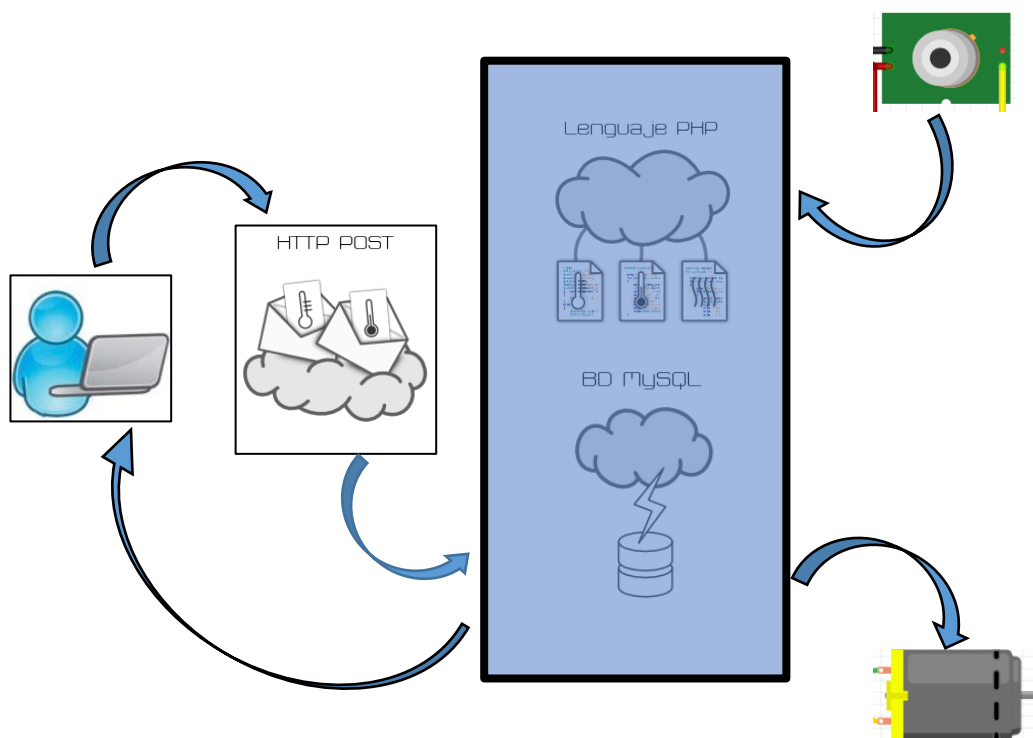


Figura 40. Arquitectura Sistema Domótico

6.3 Estructura de la Base de Datos

Diagrama Entidad Relación

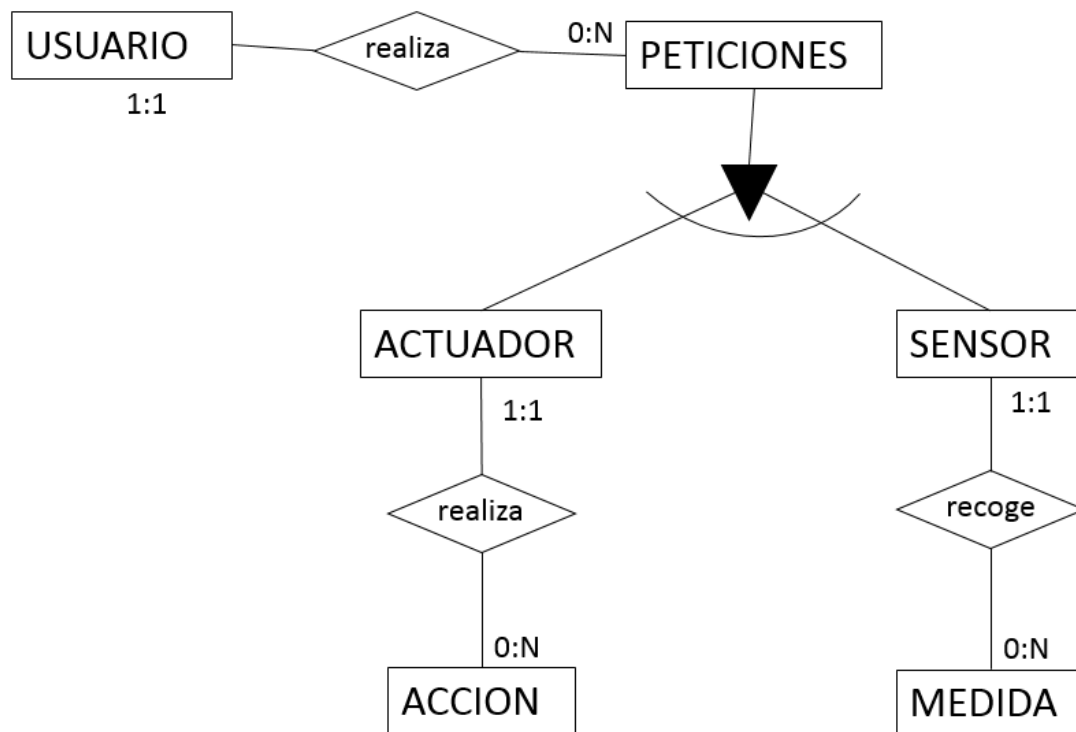


Figura 41. Diagrama Entidad-Relación

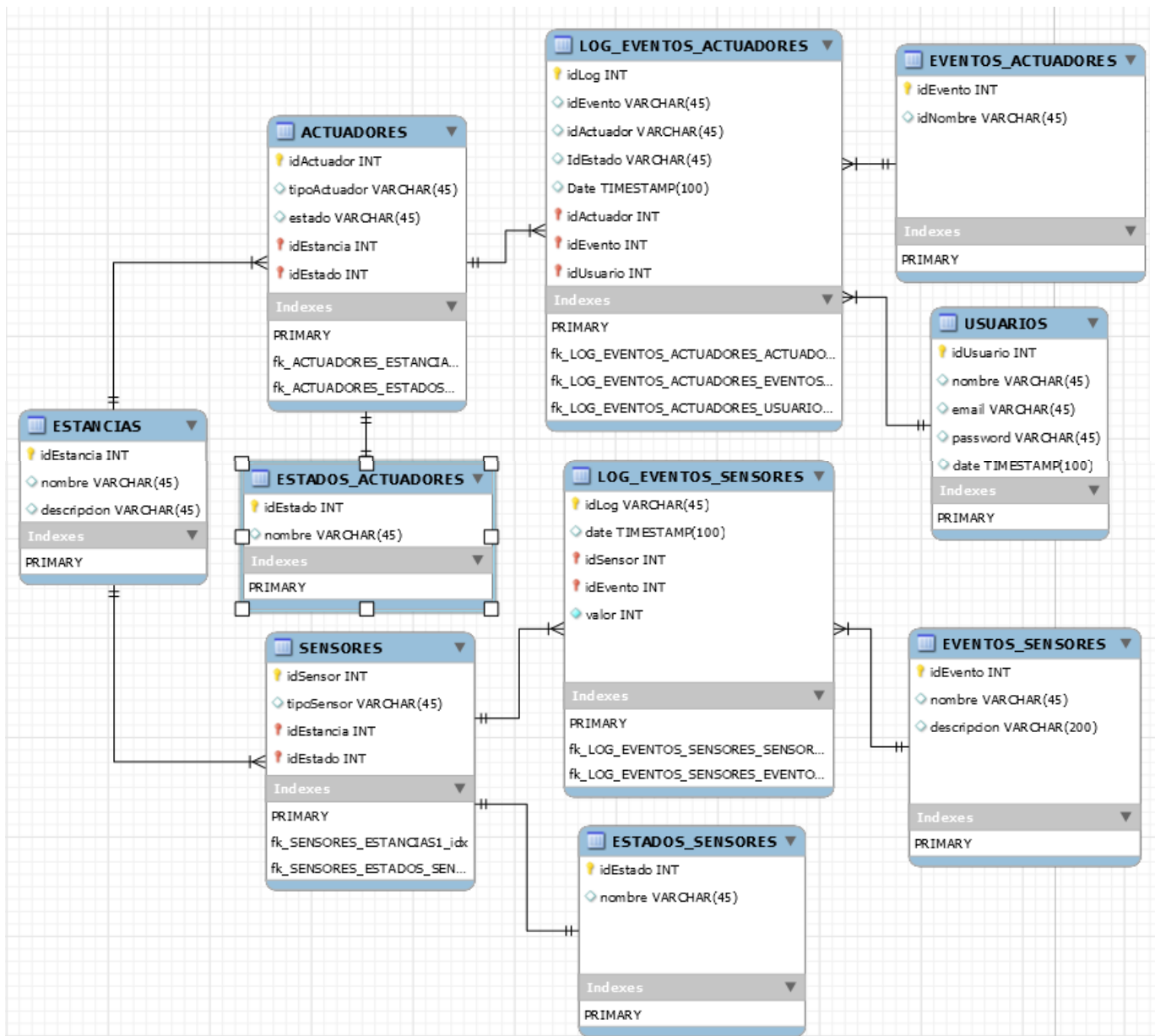
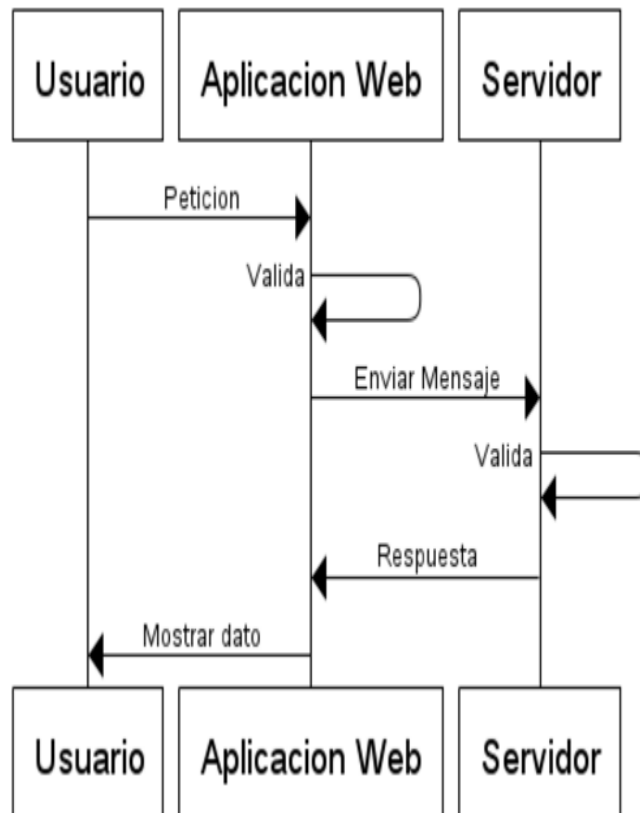


Figura 42. Diagrama Base de datos

6.4 Diseño UML del sistema

*Figura 43. Diagrama de secuencia. Validar mensajes*

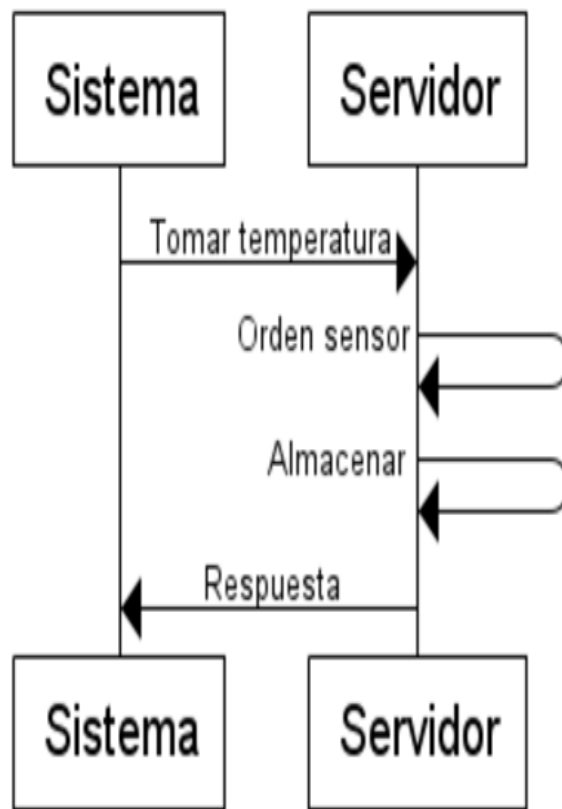


Figura 44. Diagrama de secuencia. Obtener temperatura

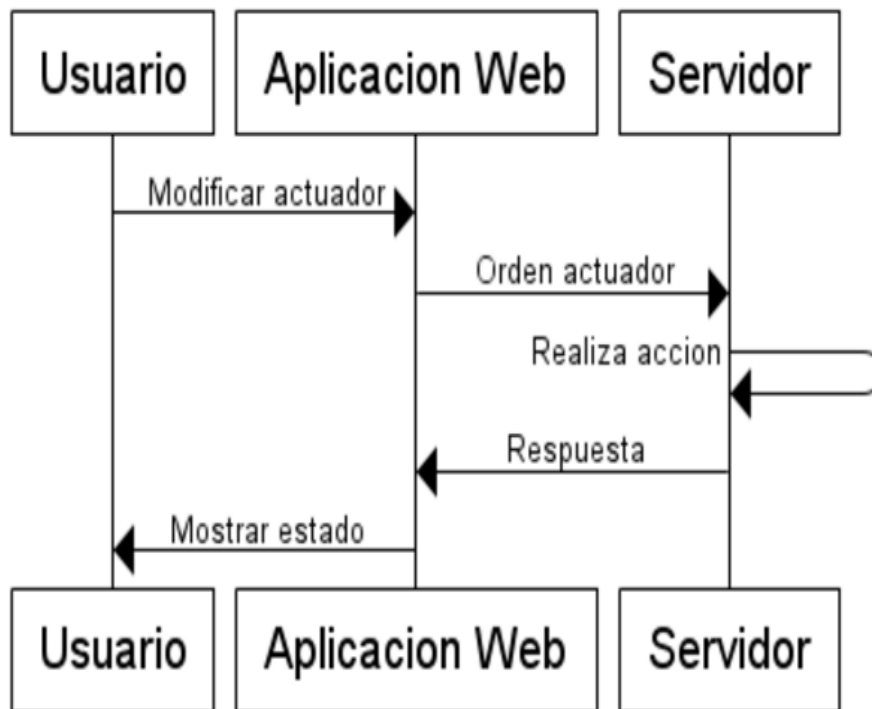


Figura 45. Diagrama de secuencia. Modificar actuador

7. Implementación del sistema

7.1 Introducción

Este apartado, tratará de aclarar al lector los métodos utilizados por el equipo de desarrollo para poder llevar a cabo el proyecto. Para ello, se especificarán las herramientas necesarias para el desarrollo de éste sistema, así como el entorno tecnológico completo utilizado.

Sin entrar a comentar ninguna línea del código de la aplicación, se tratará de realizar una breve descripción de cada una de las partes implementadas. Además, se tratará de aclarar cualquier duda respecto al desarrollo del sistema.

7.2 Herramientas de desarrollo del sistema

Para el desarrollo del proyecto se ha elegido un editor de código fuente como es Sublime Text. Este editor está escrito en C++ y Python para los plugins y aunque no es software libre como tal, la versión de evaluación es plenamente funcional y no tiene fecha de caducidad.



Figura 46. Editor Sublime Text

En la transferencia de archivos entre el micro controlador y el equipo de desarrollo se ha utilizado el software WinSCP. Este programa permite conexiones seguras mediante SFTP, así como configuraciones específicas según el tipo de servidor al que se desee conectar.



Figura 47. Logo Winscp

La comunicación entre el micro controlador y el equipo de desarrollo se ha realizado mediante el software PUTTY. La configuración de permisos, ejecución de scripts y tareas mas propias de administración de redes o base de datos, resultan de gran facilidad gracias a la consola Shell que proporciona PUTTY.



Figura 48. Logo PUTTY

A la hora de diseñar los gráficos utilizados para representar las variables físicas como temperatura, humedad, presencia y luminosidad, se ha utilizado la librería de gráficos de Google Charts. Esta librería escrita en Javascript facilita enormemente la creación de gráficos e indicadores para realizar análisis posteriores de cualquier tipo.



Figura 49. Logo Google Charts

Para el diseño de los esquemas eléctricos se ha hecho uso de la herramienta Fritzing. Este software permite la interconexión de cualquier tipo de elemento electrónico con otros elementos mediante conexiones a través de cables o protoboard estándar. Resulta de gran ayuda a la hora de diseñar y planificar los circuitos eléctricos que son necesarios en cualquier proyecto de estas características.



Figura 50. Logo Fritzing

El análisis, diseño y creación de la base de datos, ha sido llevado a cabo mediante la herramienta de desarrollo MySQL Workbench. Dicho software contiene una función de gran utilidad como es la ingeniería inversa. Dicha característica permite extraer el código SQL asociado a un esquema de base de datos, así como las relaciones existentes entre las tablas que componen el esquema.



Figura 51. Logo MySQL Workbench

8. Pruebas del sistema

En este apartado se especificarán las pruebas realizadas sobre el sistema. Se realizará una primera introducción en la que se especifique la forma de proceder para desarrollar un plan de pruebas efectivo. En dicha introducción se tratará de dar pie a la explicación de los sucesivos puntos del apartado así como exponer los casos de prueba.

8.1 Introducción

Para especificar un plan de pruebas válido y consistente, en primer lugar se expondrán los procedimientos para desarrollar las pruebas. Se destacan dos tipos de prueba. La primera, de infraestructura hardware, la segunda por funcionalidad de la aplicación.

Las pruebas de funcionalidad se dividirán en pruebas propias de la interfaz web y por otro lado las pruebas correspondientes a los elementos finales, como son los actuadores.

8.2 Procedimientos de pruebas

Se van a realizar diferentes tipos de pruebas para obtener una validación total del sistema. En primer lugar, se especificarán las pruebas de hardware, que deberán asegurar el correcto funcionamiento de toda la parte material del proyecto (sensores/actuadores), así como el sistema Raspberry pi.

Seguidamente, se presentarán las pruebas de funcionalidad de la aplicación. Estas pruebas se ejecutarán sobre el sistema que aloja la aplicación web y el servidor de la base de datos.

Para describir el conjunto de pruebas se utilizará un formato de tabla como el que se muestra en la figura X, con el siguiente contenido:

- **Identificador:** Identifica de forma unívoca cada uno de las pruebas del sistema que se explicarán a continuación. El formato para las pruebas del sistema será PXX-YY, siendo XX el tipo de prueba y YY la numeración de la misma.
- **Descripción:** Breve descripción de la prueba a realizar.
- **Requisito de Software:** Relaciona la prueba con un requisito de software ya declarado.

Identificador	PXX - YY
Descripción	
Requisito Software	
Estado inicial	
Estado final	
Procedimiento de la prueba	
Resultado de la prueba	

Tabla 57. Detalle prueba del sistema

8.2.1 Pruebas de infraestructura hardware

Prueba Infraestructura Hardware 1	PHW - 01
Descripción	Prueba de funcionamiento de micro controlador
Estado inicial	Microcontrolador desconectado
Estado final	Microcontrolador conectado
Procedimiento de la prueba	<ol style="list-style-type: none"> 1. El microcontrolador se encuentra en estado desconectado 2. Se conecta el microcontrolador 3. Se comprueba como el led indicador de encendido del microcontrolador se enciende correctamente
Resultado de la prueba	Microcontrolador conectado y funcional

Tabla 58. Prueba Hardware PHW-01

Prueba Infraestructura Hardware 2	PHW - 02
Descripción	Prueba de funcionamiento de sensor PIR
Estado inicial	Sensor PIR no envía datos
Estado final	Sensor PIR envía datos
Procedimiento de la prueba	<ol style="list-style-type: none"> 1. Se conecta microcontrolador 2. Se realiza petición de movimiento 3. El sensor devuelve la información solicitada
Resultado de la prueba	El sensor devuelve el dato de movimiento correcto

Tabla 59. Prueba Hardware PHW-02

Prueba Infraestructura Hardware 3	PHW - 03
Descripción	Prueba de funcionamiento de sensor DHT-11
Estado inicial	Sensor DHT-11 no envía datos
Estado final	Sensor DHT-11 envía datos
Procedimiento de la prueba	1. Se conecta microcontrolador 2. Se realiza petición de temperatura y humedad 3. El sensor devuelve la información solicitada
Resultado de la prueba	El sensor devuelve el dato de temperatura y humedad correcto

Tabla 60. Prueba Hardware PHW-03

Prueba Infraestructura Hardware 4	PHW - 04
Descripción	Prueba de funcionamiento de sensor LDR
Estado inicial	Sensor LDR no envía datos
Estado final	Sensor LDR envía datos
Procedimiento de la prueba	1. Se conecta microcontrolador 2. Se realiza petición de luminosidad 3. El sensor devuelve la información solicitada
Resultado de la prueba	El sensor devuelve el dato de luminosidad correcto

Tabla 61. Prueba Hardware PHW-04

Prueba Infraestructura Hardware 5	PHW - 05
Descripción	Prueba de funcionamiento de leds
Estado inicial	Led apagado
Estado final	Led encendido
Procedimiento de la prueba	1. Se conecta microcontrolador 2. Se realiza petición de modificar estado led 3. El led transita al estado solicitado
Resultado de la prueba	El led cambia al estado que se solicita desde la interfaz

Tabla 62. Prueba Hardware PHW-05

Prueba Infraestructura Hardware 6	PHW - 06
Descripción	Prueba de funcionamiento de motores
Estado inicial	Motor estado inicial
Estado final	Motor estado final
Requisito Software	
Procedimiento de la prueba	1. Se conecta microcontrolador 2. Se realiza petición de modificar estado motor 3. El motor cambia al estado solicitado
Resultado de la prueba	El motor realiza correctamente la acción solicitada

Tabla 63. Prueba Hardware PHW-06

8.2.2 Pruebas de funcionalidad

Prueba Funcionalidad Aplicación Web 1	PAW - 01
Descripción	Prueba de conexión a la aplicación web desde la misma subred donde se encuentra el servidor
Requisito Software	RSF-01
Estado inicial	Usuario sin conexión a la aplicación
Estado final	Usuario visualiza en la aplicación
Procedimiento de la prueba	1. Usuario sin identificar en la aplicacion 2. Usuario abre navegador web 3. Usuario visualiza aplicación web
Resultado de la prueba	Usuario visualiza correctamente la aplicacion

Tabla 64. Prueba Funcionalidad PAW-01

Prueba Funcionalidad Aplicación Web 2	PAW - 02
Descripción	Prueba de inicio de sesión con usuario y contraseña válidos
Requisito Software	RSF-03 RSF-04 RSF-05
Estado inicial	Usuario sin identificar a la aplicación
Estado final	Usuario identificado en la aplicación
Procedimiento de la prueba	1. Usuario sin identificar en la aplicacion 2. Usuario abre navegador web 3. Usuario visualiza aplicación web 4. Usuario introduce credenciales correctos 5. Usuario visualiza el menú principal de la aplicación
Resultado de la prueba	Usuario visualiza menú principal de la aplicacion

Tabla 65. Prueba Funcionalidad PAW-02

Prueba Funcionalidad Aplicación Web 3	PAW - 03
Descripción	Prueba de cierre de sesión
Requisito Software	
Estado inicial	
Estado final	
Procedimiento de la prueba	
Resultado de la prueba	

Tabla 66. Prueba Funcionalidad PAW-03

Prueba Funcionalidad Aplicación Web 4	PAW - 04
Descripción	Prueba de monitorización de variables ambientales
Requisito Software	RSF-06 RSF-07 RSF-08 RSF-09 RSF-10
Estado inicial	Usuario no visualiza datos ambientales
Estado final	Usuario visualiza datos ambientales
Procedimiento de la prueba	1. Usuario se conecta correctamente a la aplicación 2. Usuario visualiza menú principal 3. Usuario navega a la zona de monitorización ambiental 4. Usuario visualiza variables ambientales
Resultado de la prueba	Usuario visualiza datos ambientales

Tabla 67. Prueba Funcionalidad PAW-04

Prueba Funcionalidad Aplicación Web 5	PAW - 05
Descripción	Prueba de monitorización y control de la iluminación
Requisito Software	RSF-11
Estado inicial	Usuario no visualiza datos ambientales
Estado final	Usuario visualiza datos ambientales
Procedimiento de la prueba	1. Usuario se conecta correctamente a la aplicación 2. Usuario visualiza menú principal 3. Usuario navega a la zona de monitorización ambiental 4. Usuario visualiza variables ambientales
Resultado de la prueba	Usuario visualiza datos ambientales

Tabla 68. Prueba Funcionalidad PAW-05

Prueba Funcionalidad Aplicación Web 6	PAW - 06
Descripción	Prueba de monitorización de movimiento
Requisito Software	RSF-11 RSF-13 RSF-14 RSF-15
Estado inicial	Usuario no visualiza datos de movimiento
Estado final	Usuario visualiza datos de movimiento
Procedimiento de la prueba	<ol style="list-style-type: none"> 1. Usuario se conecta correctamente a la aplicación 2. Usuario visualiza menú principal 3. Usuario navega a la zona de monitorización de la seguridad 4. Usuario visualiza variables ambientales
Resultado de la prueba	Usuario visualiza los datos de seguridad que desea

Tabla 69. Prueba Funcionalidad PAW-06

Prueba Funcionalidad Aplicación Web 7	PAW - 07
Descripción	Prueba de monitorización de cantidad lumínica
Requisito Software	RSF-11 RSF-12
Estado inicial	Usuario no visualiza datos de iluminación
Estado final	Usuario visualiza datos de iluminación
Procedimiento de la prueba	<ol style="list-style-type: none"> 1. Usuario se conecta correctamente a la aplicación 2. Usuario visualiza menú principal 3. Usuario navega a la zona de control y monitorización de la iluminación 4. Usuario visualiza datos de iluminación
Resultado de la prueba	Usuario visualiza datos de iluminación que desea

Tabla 70. Prueba Funcionalidad PAW-07

8.2.3 Matriz de trazabilidad

	RSF-01	RSF-02	RSF-03	RSF-04	RSF-05	RSF-06	RSF-07	RSF-08	RSF-09	RSF-10	RSF-11	RSF-12	RSF-13	RSF-14	RSF-15
PAW-01	x														
PAW-02		x	x	x											
PAW-03		x	x	x	x										
PAW-04						x	x	x	x	x					
PAW-05											x	x			
PAW-06											x	x		x	x
PAW-07											x	x	x		

Tabla 71. Matriz de trazabilidad. Requisitos funcionales con pruebas de funcionalidad

9. Conclusiones y líneas futuras

9.1 Conclusiones

Tras meses de desarrollo el resultado del proyecto ha sido satisfactorio. La circunstancia de trabajar con hardware y con dispositivos sensibles como sensores o circuitos impresos, supone, en ocasiones, una gran inversión de tiempo extra. Estos tiempos se han alargado en momentos puntuales a la hora de trabajar con herramientas como el soldador o en la comprobación de corriente eléctrica en puntos determinados del circuito.

Respecto a la parte de desarrollo de software, se han profundizado conceptos adquiridos durante el transcurso de la carrera y también se han adquirido nuevos conocimientos en lo que respecta al software del control y monitorización de dispositivos.

Como conclusión final se puede decir que ha sido el trabajo más gratificante debido en gran parte a la afición personal que se ha adquirido a lo largo de los años. Con todo ello se ha interiorizado la importancia de ser capaz de desarrollar un sistema propio con recursos limitados.

9.2 Líneas futuras de mejora

El proyecto realizado se presenta finalmente como una plataforma de desarrollo domótico que permite la ampliación continua del sistema para las futuras necesidades del cliente.

La conexión a internet con la posibilidad de controlar los dispositivos de la vivienda desde cualquier lugar, es una de las mejoras que se pueden plantear.

La integración de nuevos dispositivos en el sistema resulta de gran importancia, así como las nuevas actualizaciones de software que se vayan produciendo en el futuro.

9.3 Glosario de términos

BB.DD: Bases de Datos.

SBC: Single Board Computer.

LAMP: Linux Apache MySql Php.

MVC: Modelo Vista Controlador.

GPIO: General Purpouse Input Output.

IOT: Internet of Things.

API: Application Programming Interface.

Framework: Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

HTTP: HyperText Transfer Protocol

URL: Uniform Resource Locator.

CSS: Cascading Style Sheets.

HTML: HyperText Markup Language.

PC: Personal Computer.

Wi-Fi: Wireless Fidelity.

10. Referencias

[1] PWM <http://www.knight-of-pi.org/simple-dc-motor-board-for-the-raspberry-pi-with-ic-l293-and-software-pulse-width-modulation/>

[2] https://python.swaroopch.com/about_python.html

[3] http://planner.roomsketcher.com/?ctxt=rs_com&language=es#/?pid=493174

[4] Send an email if temperature is too high
<https://www.raspberrypi.org/forums/viewtopic.php?p=672406>

[5] Controlling DC Motors Using Python With a Raspberry Pi
<http://computers.tutsplus.com/tutorials/controlling-dc-motors-using-python-with-a-raspberry-pi--cms-20051>

[6] PHP + MySQL + HighCharts: Mostrar gráfica dinámica en función del tiempo
<https://geekytheory.com/php-mysql-highcharts-mostrar-grafica-dinamica-en-funcion-del-tiempo>

[7] Raspberry Pi Temperature Sensor Web Server – Part 1 (Intro, sensor setup and Python script) <https://wingoodharry.wordpress.com/2014/12/24/raspberry-pi-temperature-sensor-web-server-part-1-intro-sensor-setup-and-python-script/>

[8] Raspberry Pi - GPIO & Python (9/9) - Passive infrared sensor
<https://learn.adafruit.com/basic-resistor-sensor-reading-on-raspberry-pi/basic-photocell-reading>

11. Anexos

11.1 ANEXO I Esquemas eléctricos

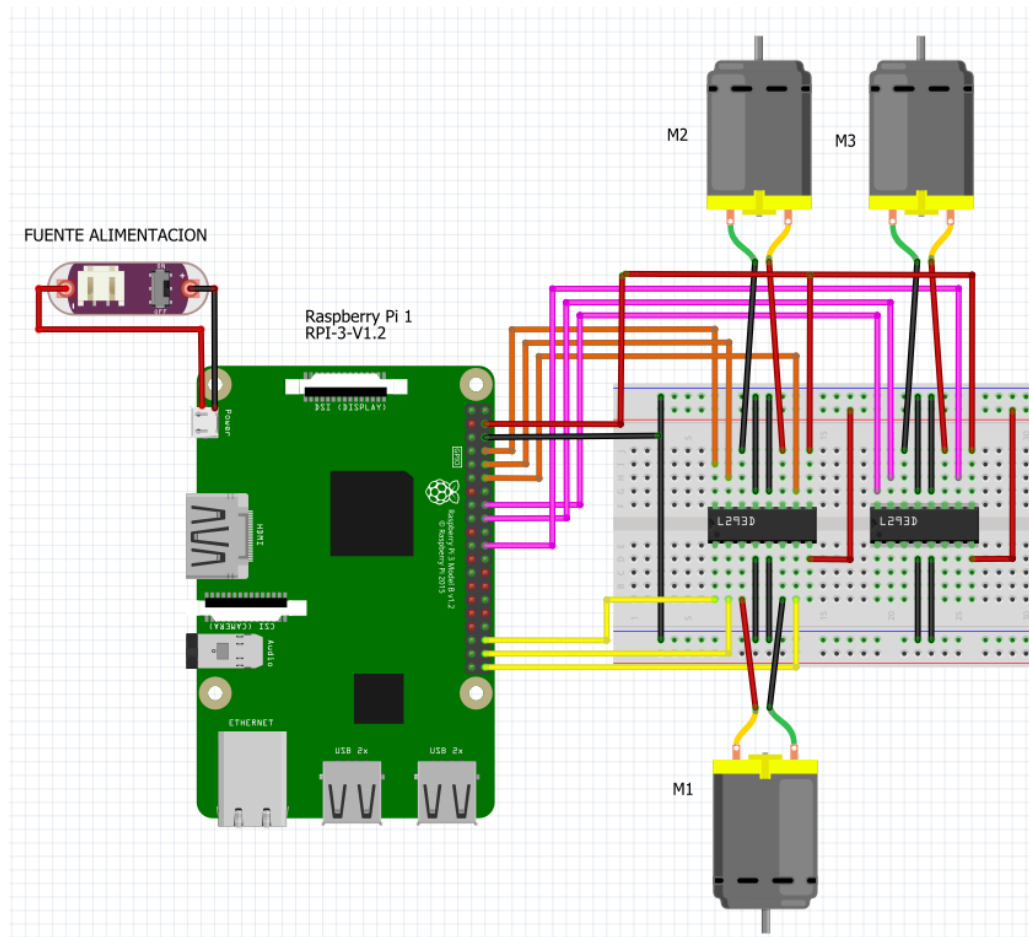


Figura 52. Esquema eléctrico motores

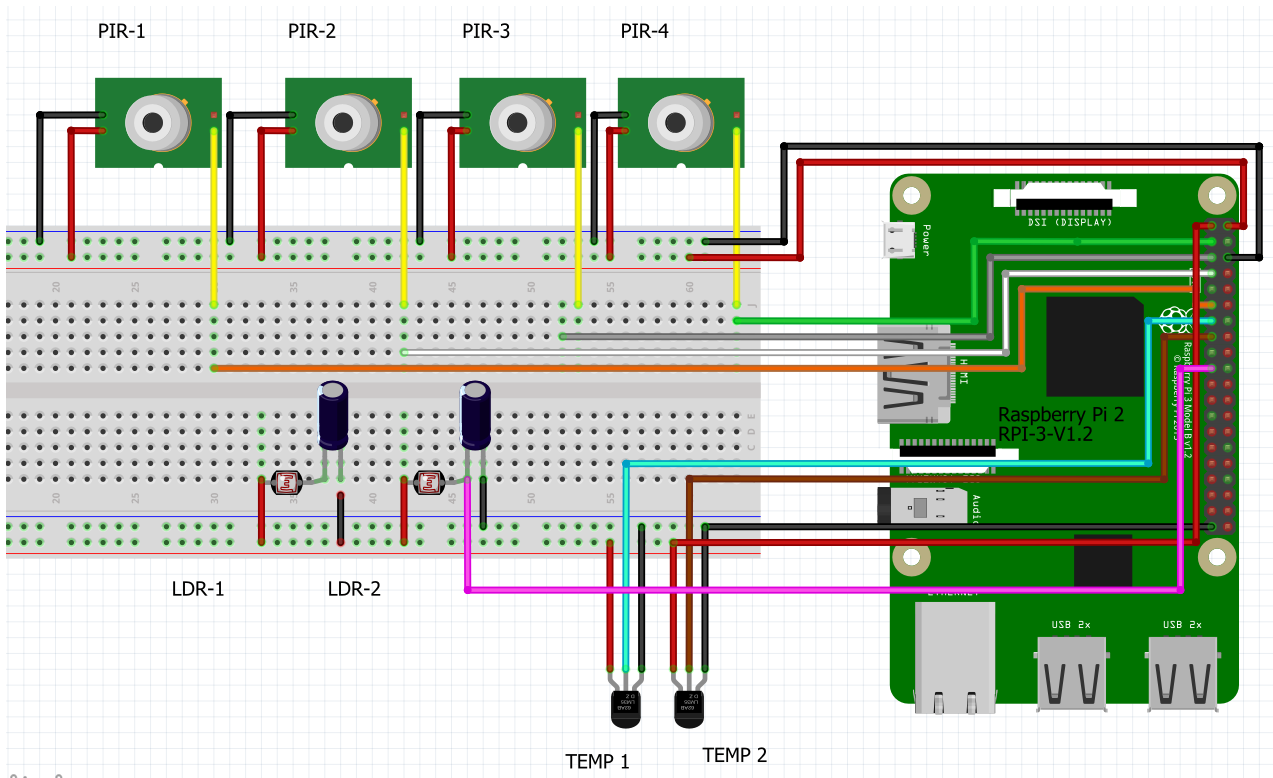


Figura 53. Esquema eléctrico sensores

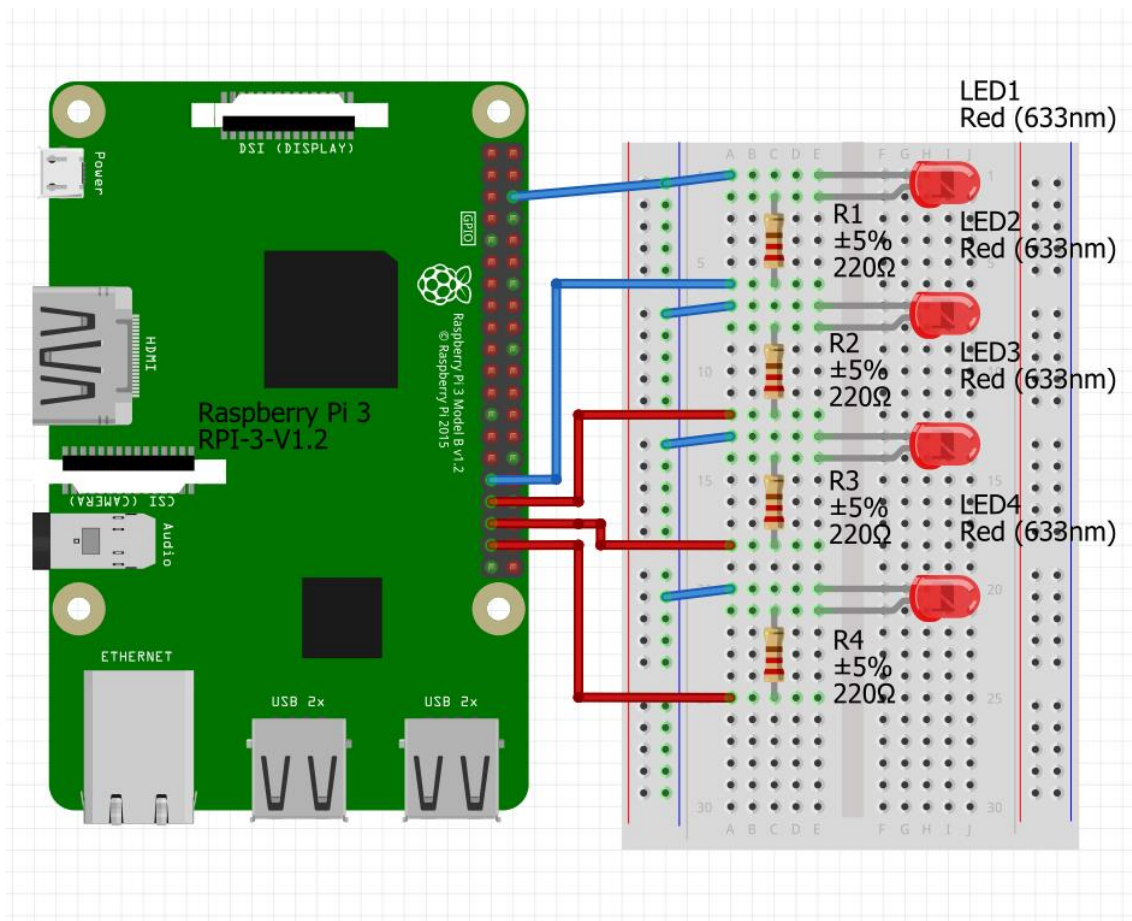


Figura 54. Esquema eléctrico leds

11.2 ANEXO II Manual de usuario

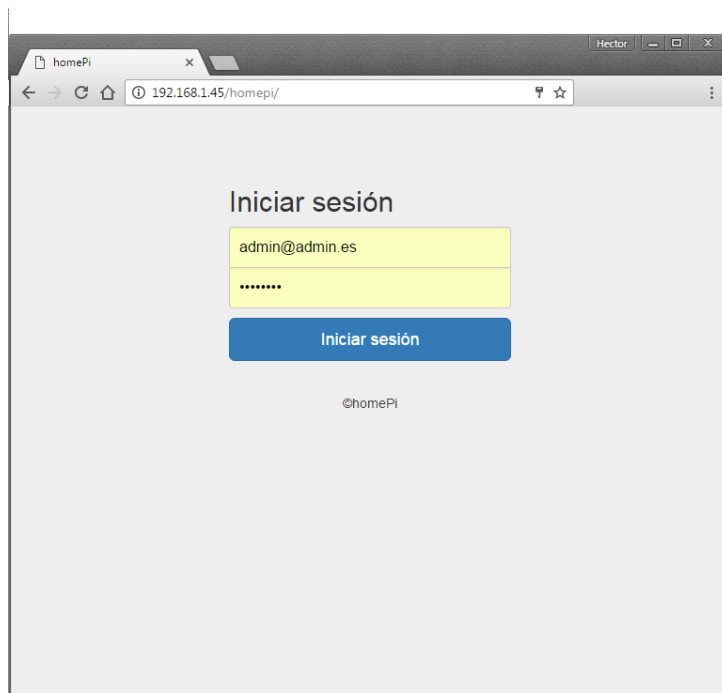


Figura 55. Pantalla login aplicación

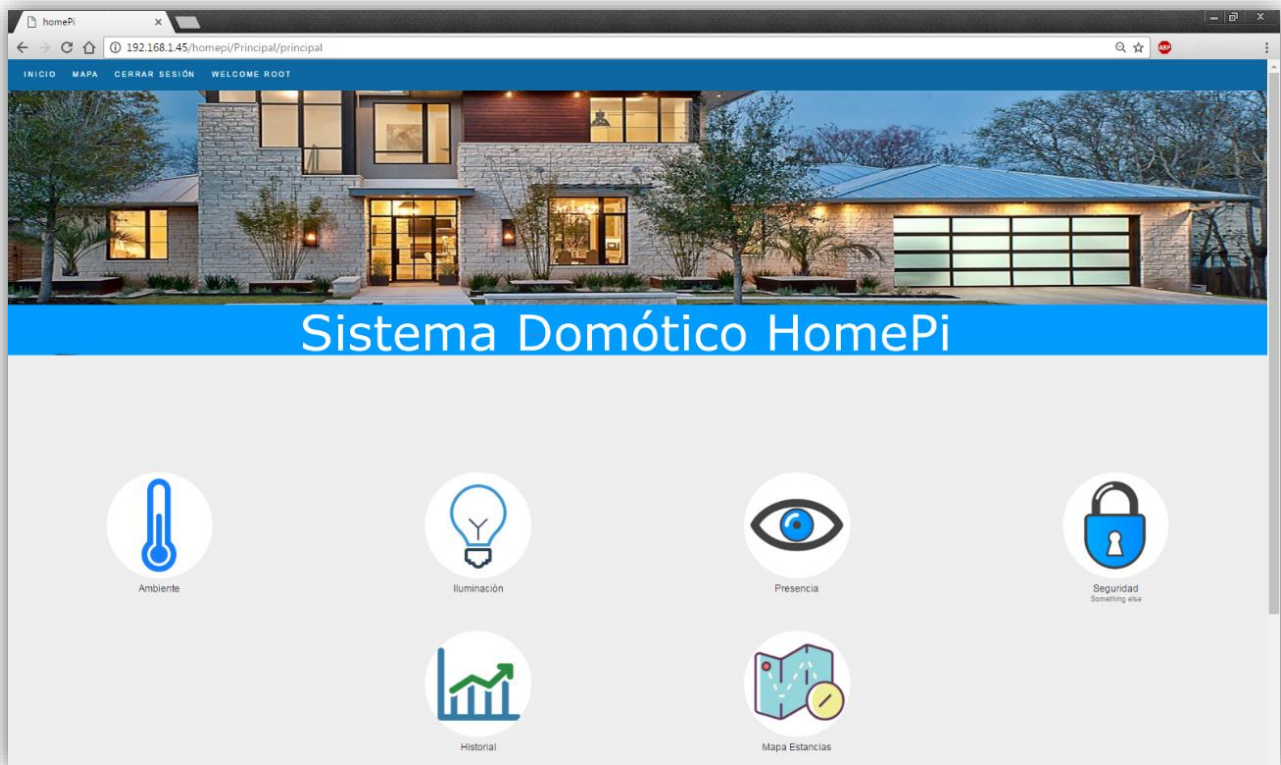


Figura 56. Pantalla menú principal

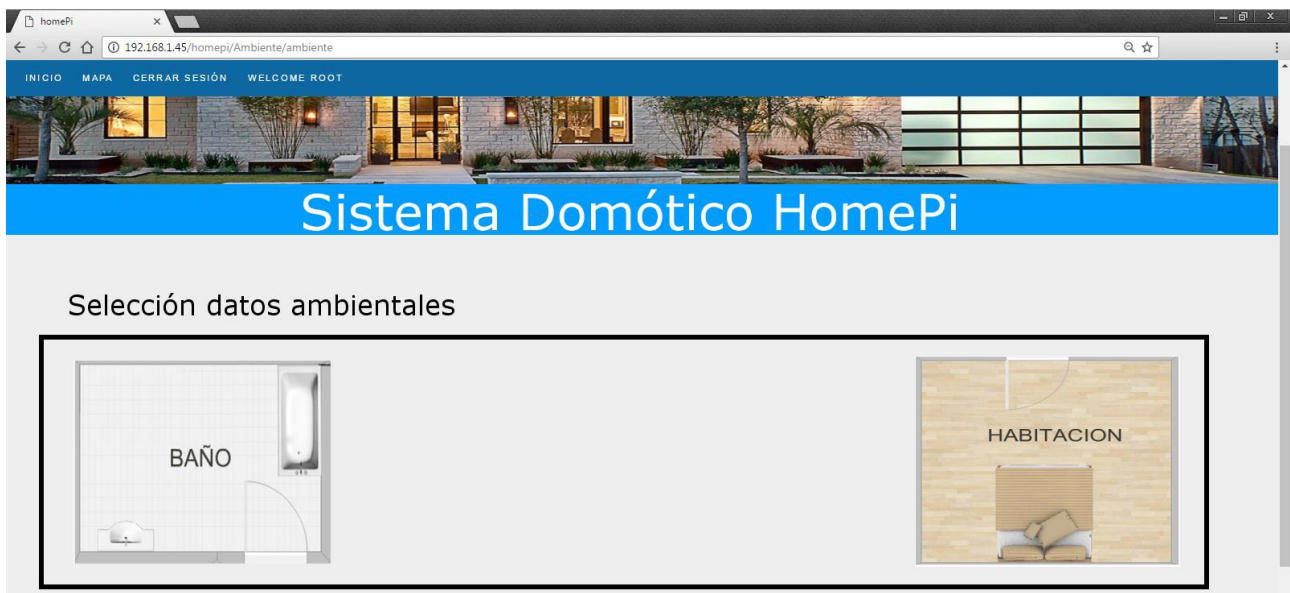


Figura 57. Pantalla selección monitorización ambiental



Figura 58. Pantalla monitorización ambiental cuarto de baño



Figura 59. Pantalla control seguridad garaje



Figura 60. Pantalla datos históricos

